Title
When do machine learning models generalize well? A signal-processing perspective

Author
Subramanian, Vignesh

Publication Date
2022

Peer reviewed|Thesis/dissertation

When do machine learning models generalize well? A signal-processing perspective

by

Vignesh Subramanian

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering – Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Anant Sahai, Chair
Professor Kannan Ramachandran
Professeor Mikhail Belkin
Professor Gireeja Ranade

Summer 2022

When do machine learning models generalize well? A signal-processing perspective

Abstract

When do machine learning models generalize well? A signal-processing perspective

by

Vignesh Subramanian

Doctor of Philosophy in Engineering – Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Anant Sahai, Chair

Contemporary machine learning systems have demonstrated tremendous success at a variety of tasks including image classification, object detection and tracking, and recommendation algorithms. This success has been driven by the enormous advances in computation capabilities that enable us to utilize big training datasets, with large number classes and train models with a vast number of parameters. In fact, these systems use models that have sufficient model capacity to be trained to zero training error on noisy or even completely random labels. However, these models often generalize well in practice and avoid harmful "overfitting". The key to good generalization lies in the implicit bias of the model architecture and training algorithm that steers us towards solutions that generalize well. This thesis works towards a better theoretical understanding of this phenomenon by analyzing overparameterized linear models and proving sufficient and necessary conditions for good generalization. Additionally, we also empirically investigate whether we can engineer the correct implicit bias when training models to solve practical problems in the field of control by making use of our knowledge about the problem domain.

We start by analyzing the simpler setting of overparameterized linear regression, fitting a linear model to noisy data when the number of features exceeds the number of training points. By taking a Fourier-theoretic perspective we map the key challenge posed by overparameterization to the well-known phenomenon of aliasing of the true signal due to under-sampling. Borrowing from the signal-processing concepts of "signal bleed" and "signal contamination", we derive conditions for good generalization for the Fourier-feature setting.

Next, we analyze the generalization error for the minimum-ℓ_2-norm interpolator for the regression and binary classification problems in a Gaussian-feature setting. For regression, we interpolate real-valued labels and for binary classification, we interpolate binary labels. (It turns out that under sufficient overparameterization, minimum-norm interpolation of binary labels is equivalent to other binary-classification training methods such as support-vector machines or gradient descent on logistic loss.) We study an asymptotic setting where the

number of features d scales with the number of training points n and both $n, d \to \infty$. Under a bi-level spiked covariance model for the features we prove the existence of an intermediate regime where we we perform well on the classification task but not on a corresponding regression task.

We then extend the analysis to the multiclass classification setting where the number of classes also scales with the number of training points, by deriving asymptotic bounds on the classification error incurred by the minimum-norm interpolator of one-hot encoded labels.

Finally, to understand how we can learn models that generalize well in practice, we empirically study the application of neural networks to learn non-linear control strategies for hard control problems where the optimal solutions are unknown and linear solutions are provably sub-optimal. By intelligently designing neural network architectures and training methods that leverage our knowledge of the dynamics of the control system, we are able to more easily and robustly learn control strategies that perform well.

To Amma, Appa and Vaishnavi.

Contents

List of Figures

List of Tables

Acknowledgments

I am grateful to be able to acknowledge my peers, advisors and family that made the five years of my PhD an enjoyable experience. First, I would like to thank my advisor Anant Sahai; Anant has been incredibly supportive through the ups and downs of research and I have learned a lot from him including the right way to tackle hard problems by adding a little bit of complexity at a time, writing research papers and responding to reviewers whose reviews can sometimes get on one's nerves. Anant has always encouraged me to pursue collaborations with others and explore research problems that I have been interested in.

Second, I would like to thank Gireeja Ranade from whom I have learned a lot not just about research but also about how to develop content for a course, interact with students, teach discussion sections and conduct homework parties. I was lucky to have been the graduate student instructor for the undergraduate optimization course twice when Gireeja was one of the instructors for the course.

I would also like to thank rest of my qualifying exam and dissertation committee members, Misha Belkin, Jiantao Jiao and Kannan Ramachandran for their valuable feedback that helped shape this dissertation.

Through the course of my PhD, I have had the pleasure of collaborating with several people, including professors, graduate students as well as undergraduate students and these collaborations have helped shed different insights into a problem. Further, these collaborations have been extremely fun since working on a problem alone can get boring sometimes. I would like to thank my co-authors Rahul Arya, Laura Brink, Daniel Hsu, Catherine Huang, Nikunj Jain, Akhil Jalan, Vidya Muthukumar, Adhyyan Narang, Josh Sanz, Nikhil Shinde, Caryn Tran, Kailas Vodrahalli and Moses Won.

I would like to thank the fellow undergraduate and graduate student instructors whom I had the opportunity of interacting with during my time as a graduate student instructor for the machine learning course in Fall 2020 and the optimization course in Spring 2019 and Spring 2020. Thanks to my friends from BLISS lab whom I have sometimes annoyed by enthusiastically (and loudly) discussing problems in the lab.

I would also like to thank my undergraduate and masters advisors Sibi Raj Pillai and Rajbabu Velmurugan that initially got me excited about research and encouraged me to pursue a PhD. Finally, I would like to thank my parents, my sister and my grandparents for their unwavering support and love.

Chapter 1

Introduction

1.1 A traditional view of machine learning

A paradigmatic problem in supervised machine learning (ML) involves predicting an output response from an input, based on patterns extracted from a possibly noisy training data set. A machine learning model is used to express this input-output relation and the goal via training is to learn the parameters for this model. Mathematically, we can express this problem as learning the model parameters $\widehat{\boldsymbol{\alpha}} \in \mathbb{R}^d$ from training data that consists of n covariate(input)-response(output) pairs $(\mathbf{X}_i, Y_i)_{i=1}^n$. The model takes $\mathbf{X}_i$ as input and predicts the response $\widehat{Y}_i = f(\mathbf{X}_i; \widehat{\boldsymbol{\alpha}})$, which is a function of the model parameters. Typically, we learn the model parameters $\widehat{\boldsymbol{\alpha}}$ by minimizing the training loss (also called empirical risk minimization) between the predicted and true responses as illustrated in Figure 1.1,

$$\widehat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \sum_{i=1}^{n} \ell_{\text{train}} \left(f(\mathbf{X}_i; \widehat{\boldsymbol{\alpha}}), Y_i \right). \tag{1.1}$$

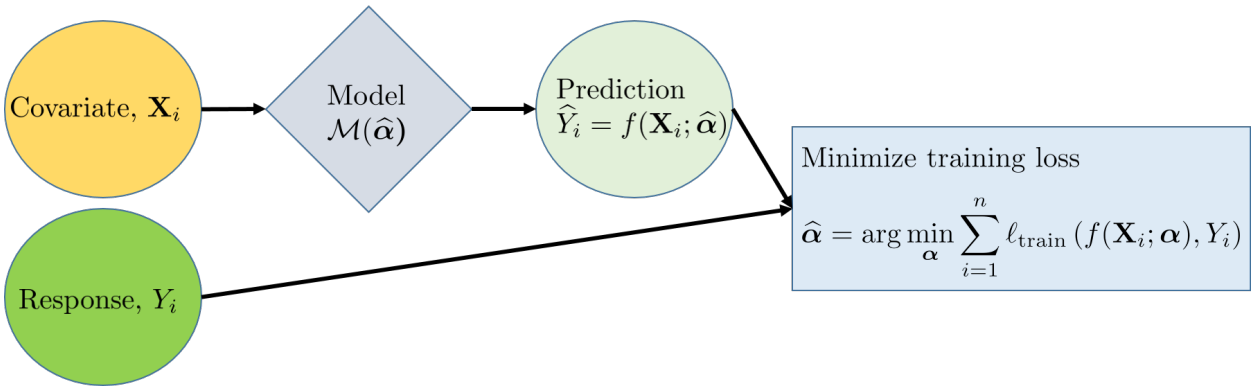


Figure 1.1. Illustration of training a supervised machine learning model via empirical risk minimization.

We are interested in the performance of our model on a separate test or validation data set

$$\mathcal{E}_{\text{test}} = \mathbb{E}_{(\mathbf{X},Y)\sim P}\ell_{\text{test}}\left(f(\mathbf{X};\widehat{\boldsymbol{\alpha}}),Y\right), \tag{1.2}$$

where the test point is drawn from the test distribution P. Typically, the training points are assumed to come from this same distribution but have some additional noise present. We say the model generalizes well if it has low test error, i.e. we are able to correctly predict a response on data that has not been seen before during training.

Traditionally, there are two major fears while learning the model parameters that might lead to poor generalization. First, the model is not expressive or rich enough, i.e. it lacks the model capacity to accurately transform the input into the output causing us to "underfit" the data. Second, our model might be too expressive, i.e. the model capacity is too high, causing us to "overfit" to the noise present in the training data resulting in poor performance on unseen test data. Example 1 illustrates these two scenarios in the simple setting of function approximation with polynomials.

Example 1. ***Function approximation with polynomials*** *Suppose we have $n = 16$ training covariate-response pairs (X_i, Y_i) where,*

$$Y_i = X_i^4 - X_i^2 + 1 + \varepsilon_i, \tag{1.3}$$

where $\varepsilon_i \sim \mathcal{N}(0, 1.6 \times 10^{-3})$ is additive white Gaussian noise and $X_i \sim \text{Unif}[-1, 1]$. Thus our data comes from a degree-4 polynomial function but is corrupted by noise.

Consider a polynomial model that consists of d parameters, $\widehat{\boldsymbol{\alpha}} \in \mathbb{R}^d$. On a point X_i we predict the response as,

$$\widehat{Y}_i = f(X_i;\widehat{\boldsymbol{\alpha}}) = \sum_{j=1}^{d} \widehat{\alpha}_{j+1} X_i^{j-1}.$$

We learn the model parameters by minimizing the squared loss between the predicted and labelled responses on the training data,

$$\widehat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\text{argmin}} \sum_{i=1}^{n} \left(\widehat{Y}_i - Y_i\right)^2. \tag{1.4}$$

Here, when $d \leq n$, we have a unique minimizer. However for $d > n$, there are infinitely many solutions that achieve zero training error. But not all will be a good approximation of the true function on test data points. In Figure 1.2 we consider one particular solution, the minimum-ℓ_2-norm interpolator which corresponds to running gradient descent on the squared loss in (1.4) starting from a zero initialization.

We are interested in how well we can approximate the true function, i.e. given test data (X, Y) that is generated as $Y = X^4 - X^2 + 1$, how close is our prediction $\widehat{Y}$ to the true value.

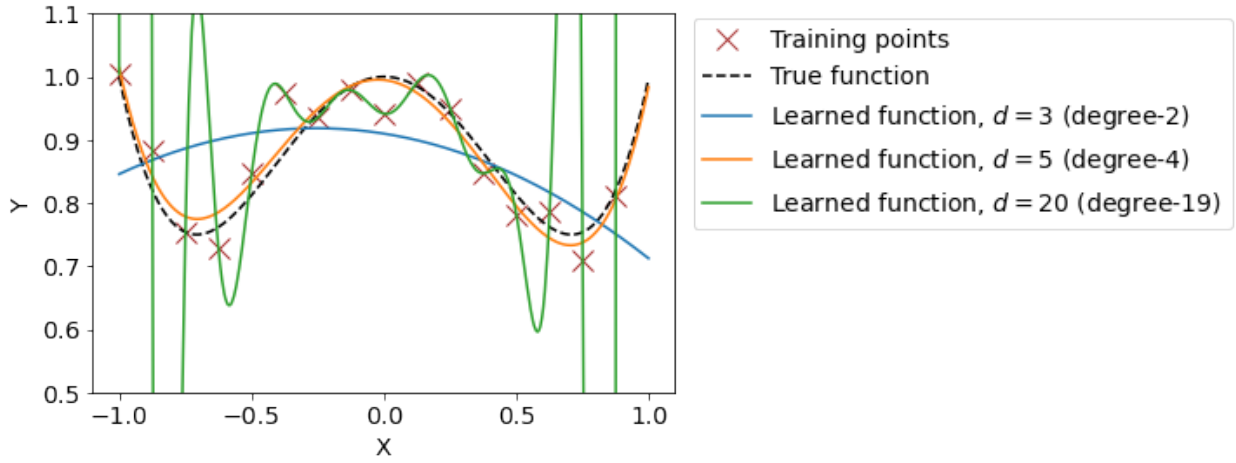


Figure 1.2. Illustration of underfitting and overfitting while performing function approximation using polynomials. The true function is shown via dotted lines in black. The $n = 16$ training points come from the true function but are corrupted by additive white Gaussian noise with zero mean and variance 1.6×10^{-3}. If we fit a degree-2 polynomial as shown in blue, we underfit the data, whereas if we fit a degree-19 polynomial (using minimum ℓ_2−-norm interpolation) we overfit the data as shown in green. Fitting a degree-4 polynomial leads to a good approximation of the true function as shown in orange.

Now, from (1.3) *we know that the optimal choice of d is 5 since the underlying true function is of degree-4. If we choose a smaller value, say* $d = 3$ *then our model is not expressive enough to capture the true function and we underfit the data. On the other hand, if we choose a larger value, say* $d = 20$ *then we can end up learning a way-too-complex function and overfit the data. Figure 1.2 illustrates these different scenarios.*

Note that the ML problem described above is related to the signal-processing problem of reconstructing a continuous time signal from noisy evenly spaced discrete samples. There again, since there can be infinitely many continuous time signals corresponding to a given set of discrete samples we need additional knowledge/assumptions about the signal to reconstruct the signal accurately. For instance, if the true signal is band-limited then we can reconstruct the signal using an ideal low pass filter, i.e. sinc interpolation in time domain though in practice we perform interpolation using raised cosine filters. The problem of non-uniqueness of reconstruction and discovery of aliases in the signal-processing domain is related to the problem of overparameterization in ML and we elaborate more on this in Section 2.2.

1.2 A contemporary view of machine learning

Contemporary machine learning systems have been used to great success to perform a variety of tasks such as image classification, object detection for identifying obstacles in autonomous

driving systems, and online recommendation algorithms for movies, songs, videos, etc. The underlying input-output relationship that the models used by these systems capture are highly complex and the models themselves are gigantic with parameters that vastly exceed the (also large) number of data points used to train these models. Such big models eliminate the risk of underfitting but however conventional wisdom is that when the number of parameters is more than the number of training data points it leads to poor generalization due to overfitting.[1]

However, in defiance of conventional wisdom regarding overfitting, these big models that can be trained to achieve zero training error even with noisy labels, still generalize well in practice [158, 53]. How can this happen? There has been a prolific amount of research in the past few years that builds towards a better theoretical understanding of this phenomenon.

1.3 Outline of thesis

In the first part of this thesis (Chapters 2,3,4), we study generalization for overparameterized *linear* models. While linear models are much simpler than the deep network models that are commonly used in practice nevertheless they are still a rich enough class to study and understand various phenomenon related to overparameterized learning and gain insights into what happens when using more complex models. The input-output relation of linear models can be expressed as,

$$Y = f(\mathbf{X}; \widehat{\boldsymbol{\alpha}}) = \phi(\mathbf{X})^\top \widehat{\boldsymbol{\alpha}},$$

where $\widehat{\boldsymbol{\alpha}}$ are the model parameters and ϕ is a featurization map. Thus, while our model is linear in the feature space it can be non-linear in the underlying covariates $\mathbf{X}$. For these generalized linear models the training procedure (1.1) and relevant test error (1.2) can be expressed as:

$$\widehat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \sum_{i=1}^{n} \ell_{\text{train}} \left(\phi(\mathbf{X}_i)^\top \boldsymbol{\alpha}, Y_i \right),$$

$$\mathcal{E}_{\text{test}} = \mathbb{E}_{(\mathbf{X},Y)\sim P} \ell_{\text{test}} \left(\phi(\mathbf{X})^\top \widehat{\boldsymbol{\alpha}}, Y \right).$$

We consider three problems. First is the regression problem where the Y_i are real-valued and both the train and test loss are the squared loss. In an overparameterized setting, although there are infinitely many choices of parameters $\widehat{\boldsymbol{\alpha}}$ that can perfectly fit or interpolate the noisy training data, we centre our analysis on the minimum-ℓ_2-norm interpolating solution since gradient descent on squared loss converges to this particular solution when initialized at zero([45]). Chapter 2 provides a signal-processing perspective inspired analysis of

[1] This wisdom is corroborated in theory by worst-case generalization bounds on such overparameterized models following from VC-theory in classification [143] and ill-conditioning in least-squares regression [98].

the minimum-ℓ_2-norm interpolation where we map overparameterization to the well known phenomenon of aliasing. The challenge while performing minimum-ℓ_2-norm interpolation is that the aliases look identical to the true signal on training data points and thus the training algorithm cannot distinguish between the two. As a result, the recovered signal has contribution from both the true signal and its aliases. Consequently, there is shrinkage of the true signal, which we denote as *signal bleed*. Further, these falsely discovered aliases differ from the true signal on test points and thus contaminate our prediction, Using a Fourier toy model, where we have regularly spaced training data points and Fourier features, we are able to quantify the challenge posed by aliasing by introducing two key quantities of interest, *survival*, a measure of how much of the true signal is recovered and *contamination*, a measure of how much contamination is introduced by the discovery of the aliases. Figure 1.3 provides an illustration of the aliasing phenomenon.

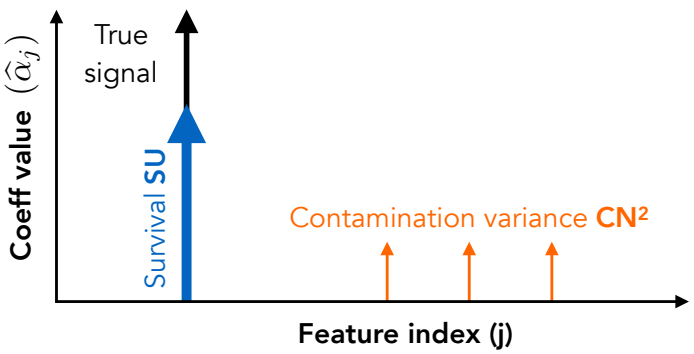


Figure 1.3. Depiction of signal components "bleeding" out into spurious features as a result of using the minimum-ℓ_2-norm interpolator. The "bleeding" has two effects: lower "survival" of signal in the original true features, and higher "contamination" by spurious features.

This chapter is based on the paper [105] and is joint work with Vidya Muthukumar, Kailas Vodrahalli and Anant Sahai.

The second problem is that of binary classification where the Y_i are binary valued (+1, -1) and here the training loss is traditionally the logistic loss or the hinge loss while the test-loss is the 0-1 loss based on whether we predicting the class correctly. However, empirically it has been observed that training with logistic or hinge loss has comparable performance to training with squared loss [122], [63]. In our work, we study the binary classifier obtained by minimum-ℓ_2-norm interpolation of the binary labels. In sufficiently overparameterized settings this classifier is identical to the one obtained by other training methods like the max-margin SVM (minimization of hinge loss) or minimization of logistic loss [104, 62]. In Chapter 3 we highlight the difference between the binary classification and regression problem by answering the key question "Is binary classification easier than regression?" To answer this question we study an asymptotic setting, where the number of training points goes to infinity and the number of features scales with the number of training points. Further, in

our model the underlying[2] covariance matrix of the features has a bi-level structure with a few large eigenvalues and several small eigenvalues as shown in Figure 1.4.

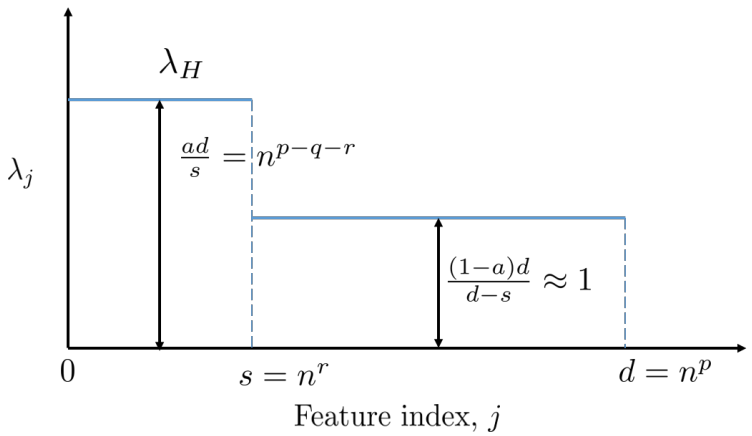


Figure 1.4. The bi-level model parameterized by p, q, r that scales with number of training points n. The number of features $d = n^p$. The covariance matrix has a bi-level structure with the first $s = n^r$ eigenvalues having value n^{p-q-r} while the remaining eigenvalues are approximately 1.

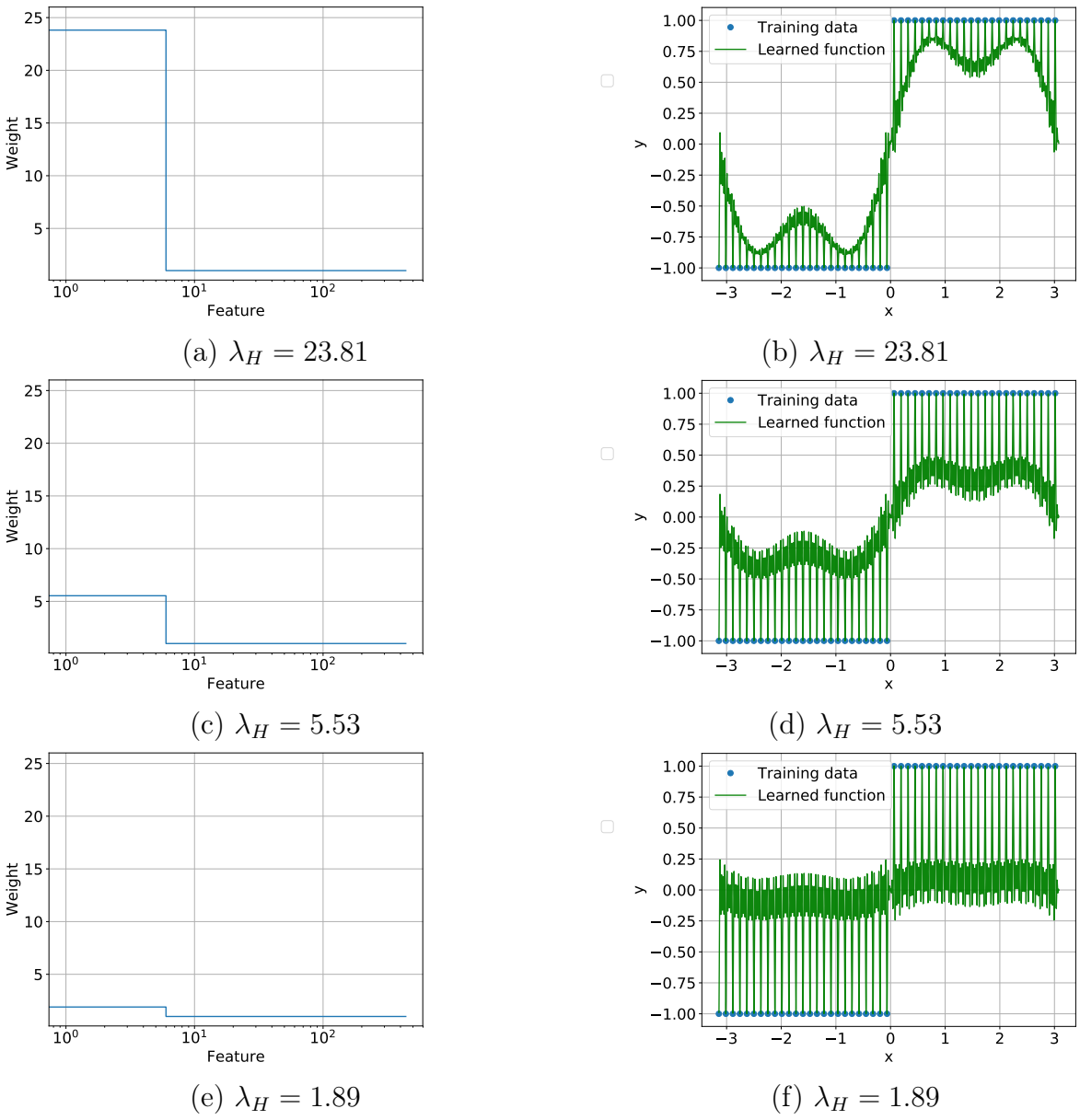
Here, experiments in a Fourier-feature setting provided empirical evidence of the existence of an intermediate regime of special interest where the minimum-ℓ_2-norm interpolator generalizes poorly in regression tasks, but well in binary classification tasks as shown in Figure 1.5.

Subsequent theoretical analysis of a Gaussian-features setting with bi-level model proved the existence of this intermediate regime beyond Fourier features as visualized in Figure 1.6.

In the language of survival-contamination, for regression to generalize well survival must go to 1 and contamination must go to zero. However, for binary classification to generalize well we only require that survival is strong enough to overcome contamination, i.e the ratio survival/contamination (this plays a similar role to the signal-to-noise ratio) goes to infinity. This chapter is based on the paper [104] and is joint work with Adhyyan Narang, Vidya Muthukumar, Misha Belkin, Daniel Hsu and Anant Sahai.

The third problem is that of multiclass classification and is studied in Chapter 4. Most practical real-world problems involve more than two classes so it is natural to study what happens when we moved beyond the binary setting. In the multiclass setting, we have k classes and the Y_i are categorical labels in the range 1 to k and the training loss is typically the cross-entropy loss while the test-loss is the $0-1$ loss. In our work we study the minimum-ℓ_2-norm interpolator of one-hot encoded labels (i.e. training loss is the squared loss) and make use of the equivalence between different choice of training loss functions from [146]. We study an asymptotic setting as before where now in addition to the number of features and the bi-level covariance model scaling with the number of points, the number of classes

[2]Section 4.7 highlights the difference between the underlying covariance matrix and the empirical eigenstructure of our feature matrix.



(a) $\lambda_H = 23.81$ (b) $\lambda_H = 23.81$

(c) $\lambda_H = 5.53$ (d) $\lambda_H = 5.53$

(e) $\lambda_H = 1.89$ (f) $\lambda_H = 1.89$

Figure 1.5. The three qualitative regimes illustrated using Fourier features and regularly spaced training points. The top corresponds to both regression and classification succeeding, the middle one is the intermediate regime where only classification works, and the bottom one is where neither works. Here $n = 49, s = 7, d = 441$.

also scales with the number of points. When is good generalization possible in the multiclass setting and how much harder is multiclass classification compared to binary classification? It turns out that, as was the case in binary classification, the ratio of the relevant survival to contamination terms plays the role of the effective signal-to-noise ratio and shows up as a key quantity in our error analysis. Asymptotically, when this ratio grows to infinity, multiclass

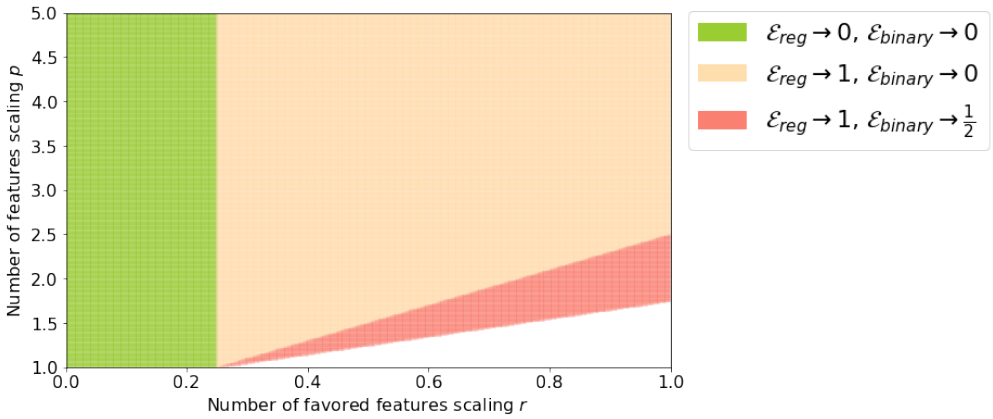


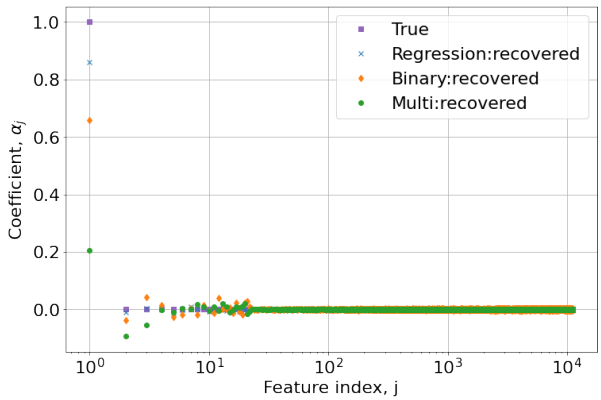
Figure 1.6. Visualization of the three asymptotic regimes from Theorem 2 for $q = 0.75$. In the green region both regression and classification generalize well and in the red region neither regression nor classification generalize well. There is an intermediate region shown in orange where classification generalizes even though regression does not.

classification generalizes well. The key additional challenge when working with multiclass training data is that there are relatively fewer positive examples of each class which results in a factor k drop in survival but only a factor $\sqrt{k}$ drop in contamination and this effectively makes multiclass classification "harder" than binary classification. Consequently, there is a limit to how many classes we can handle in the multiclass setting. This chapter is based on the preprint [133] and is joint work with Rahul Arya and Anant Sahai.

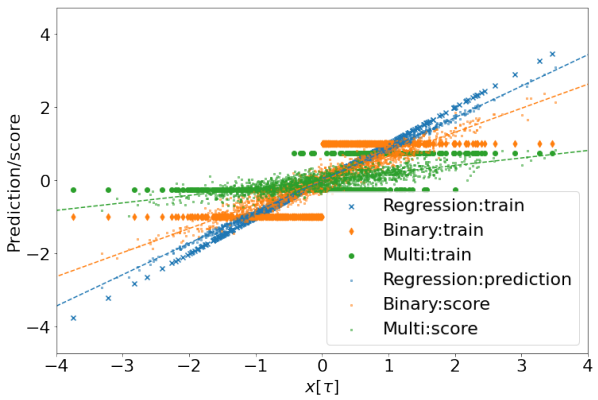
Chapters 2, 3 and 4 together highlight the key differences between the regression, binary classification and multiclass classification problems. On the one hand, the training data being less informative i.e. real-valued for regression, binary valued for binary classification and one-hot for multiclass classification leads to lower survival (fraction of signal that is recovered). Figure 1.7 visualizes this difference when we perform minimum-ℓ_2-norm interpolation in a Gaussian-feature setting under the bi-level covariance model of Figure 1.4. On the other hand, the classification task itself being easier than the regression task offsets the drop in survival and consequently classification can succeed even when regression fails. Since Chapters 2, 3 and 4 of this thesis build towards one big story, we present all relevant related work in Section 1.4 below. Additionally, there are several excellent surveys in this area [11, 13, 38] that we recommend.

The second part of the thesis, Chapter 5, empirically explores how we can engineer the right kind of implicit bias via the choice of neural network architectures and training procedures while learning non-linear control strategies for control problems. We study two control problems where linear control strategies are provably sub-optimal.

First, we study the Witsenhausen problem, a simple decentralized stochastic control problem with two controllers as illustrated in Figure 1.8. The first controller receives X_0 as input where X_0 is a zero-centered Gaussian random variable with variance σ_x^2. Observing X_0 perfectly, the first controller determines the control U_1 and the state evolves to be $X_1 =$



(a) Comparison of recovered coefficients.



(b) Comparison of the learned predictor/scores. The dashed lines are the trend lines corresponding to $y = \alpha[\tau]x$, i.e. they have slope corresponding to the recovered coefficient at the true feature index.

Figure 1.7. Visualization of the difference in survival while using the minimum-ℓ_2-norm interpolator in the regression, binary classification and multiclass classification problems. We assume a bi-level covariance model with $n = 500, d = 11181, s = 23, a = 0.21$ and $k = 4$. In subfigure (a) we plot the recovered coefficients obtained by interpolation of real-valued labels (for regression), binary labels (for binary classification) and one-hot labels (for multiclass classification). The true coefficient corresponding to the underlying true function that generated the data is 1-sparse. Subfigure (a) shows the difference in survival, i.e. fraction of recovered signal for the three cases. Subfigure (b) plots the learned predictor (for regression) and the score (for binary classification and multiclass classification) versus the value of the true feature (i.e the feature corresponding to index τ, where true coefficient is 1). We can see the effect of the 3 different types of training data. For regression training data is simply along the line with slope 1, for binary classification the training label is 1 or -1 depending on sign of the $x[\tau]$ while for multiclass classification the training label is positive only when $x[\tau]$ is largest of the first k features. Looking at the recovered predictor/scores we see the difference in slopes for the three problems. Corresponding to regression the slope of the predictor is close to 1, i.e survival is high, for binary classification the slope of the score function is smaller while for multiclass classification the slope drops even further.

$X_0 + U_1$. The second controller then receives a noisy version of the state, $Y_2 = X_1 + Z$, where Z is a standard unit variance normal random variable. Given Y_2, the second controller determines the control U_2 and the final state evolves to be $X_2 = X_1 - U_2$. The controllers are designed together to ideally minimize the expected cost function $k^2\mathbb{E}[\|U_1\|^2] + \mathbb{E}[\|X_2\|^2]$.

The two parameters σ_x^2 (how variable is the initial state) and k^2 (how heavily we penalize the first controller's control input) define an instance of the Witsenhausen problem. It is well known that the optimal control strategy for the second controller is to output the conditional expectation of X_1 given Y_2, $\mathbb{E}[X_1 \mid Y_2]$, however the optimal control strategy for the first controller is more challenging to determine. An interesting feature about the the

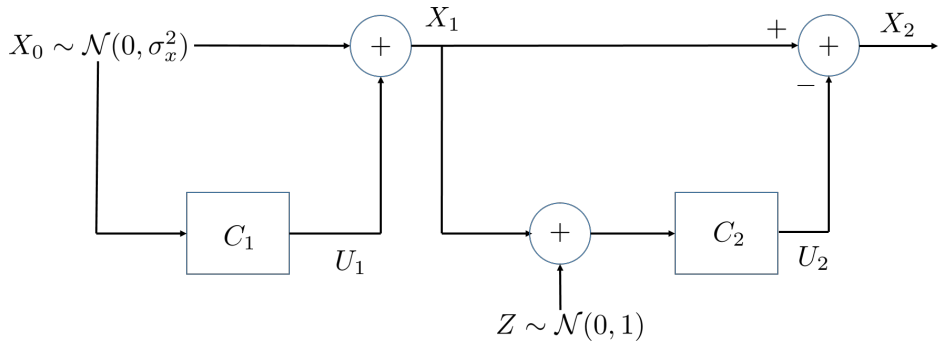


Figure 1.8. The Witsenhausen problem. The objective is to minimize $k^2\mathbb{E}[\|U_1\|^2] + \mathbb{E}[\|X_2\|^2]$.

Witsenhausen problem is that linear control strategies are provably sub-optimal [150]. In our work, we explore whether we can use neural networks to learn a non-linear control strategy for the first controller while using the conditional expectation for the second controller. Inspired by Chapters 2,3 and 4 where the implicit bias of the training data/model and the training algorithm was crucial for ensuring good generalization we use our knowledge of the problem domain and introduce a "lattice layer" in the neural networks helps us bias our network to learn a slopey-quantization-like strategy that perform well. The work on the Witsenhausen problem is based on [132] and is joint work with Laura Brink, Nikunj Jain, Kailas Vodrahalli, Akhil Jalan, Nikhil Shinde and Anant Sahai.

Second, we investigate whether a similar idea of intelligently choosing network architecture to imbue the correct implicit bias can help us tackle more complex infinite horizon control problems. We study the problem of stabilizing a linear control system with multiplicative noise where the system dynamics are given by:

$$X_{n+1} = aX_n - U_n \tag{1.5}$$

$$Y_n = Z_n X_n. \tag{1.6}$$

Here, X_n is the state of the system. The controller may choose the control (U_n) based on an observation (Y_n) that is corrupted by multiplicative noise $(Z_n \sim \mathcal{N}(0, 1))$. The goal is to ensure that the system is stable in second-moment sense, i.e. $\sup_n \mathbb{E}[X_n^2] < \infty$.

It is notable that the optimal linear strategy for this system is $U_n = 0$ for all n, however non-linear strategies can significantly (and unboundedly) improve on the performance of the linear strategy [42]. A key observation in [42] was that using the controller's memory, i.e. at time n using not only the value of Y_n but also the values of Y_{n-1}, Y_{n-2}, etc. to non-linearly generate U_n, improved the controller performance. We build on this idea and use neural networks and use the memory of multiple observations to design controllers for this seemingly simple but still-open control problem. The challenge here is that we can only learn our control strategy by training up to a finite-training horizon. However, our control strategy must generalize well, in the sense that is should continue to stabilize the system beyond the training-horizon. By choosing a periodic control structure and a greedy

training procedure coupled with input-output scaling across time, we are able to learn control strategies that generalize to time-horizons well beyond the finite training-horizon. The work on the multiplicative noise control system is based on the paper [134] and is joint work with Moses Won and Gireeja Ranade. Related work for this chapter is provided in Section 1.5 of this introductory chapter.

1.4 Related work on overparameterized learning

Double descent phenomenon

Classically, by either operating in the underparameterized regime or by performing explicit regularization, we can force the training procedure to average out the harmful effects of training noise and thereby hope to obtain good generalization. The present cycle of seeking a deeper understanding began after it was observed that modern deep networks were over-parameterized, capable of memorizing noise, and yet still generalized well, even when they were trained without explicit regularization [112, 158]. Experiments in [53, 14] observed a double-descent behavior of the generalization error where in addition to the traditional U-shaped curve in the underparameterized regime, the error decreases in the overparameterized regime as we increase the number of model parameters. Figure 1.9 from [14] reproduced here for the convenience of the reader illustrates this double-descent phenomenon.

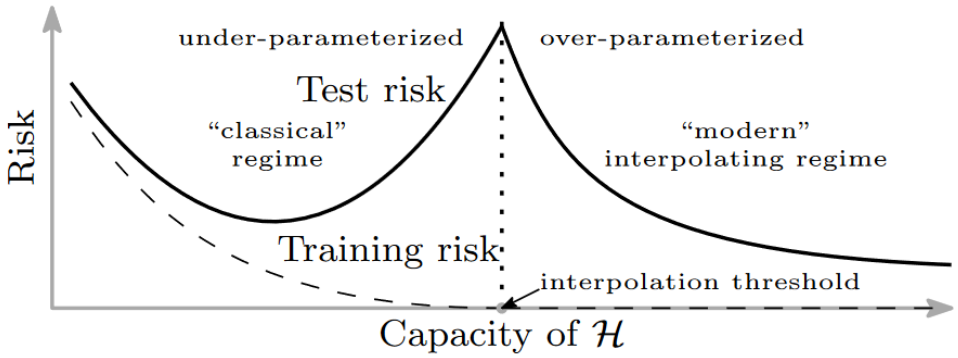


Figure 1.9. The double-descent curve for test risk. The test risk/generalization error exhibits a U-shaped behavior in the underparamterized regime. When the model capacity is too low, the test risk is high because we end up "underfitting" the data while when model capacity is too large, the test risk is high because we end up "overfitting" the data by learning an overly-complex model. However, in the overparameterized regime when the model capacity is larege enough that interpolation (zero training risk) is possible, sometimes the test risk undergoes a second descent. (Figure reproduced from [14].)

This double-descent phenomenon is not unique to deep learning models and was replicated for kernel learning [16]. Further, the good generalization performance in the overparameterized regime cannot be explained by traditional worst-case generalization bounds based on Rademacher complexity or VC-dimension since the models have the capacity to fit purely

random labels. Overparameterized models must therefore have some fortuitous combination of the model architecture with the training algorithm that leads us to a particular solution that generalizes well.

Implicit bias and role of training loss function

In the overparameterized regime, there are infinitely many solutions that interpolate training data (we saw an example of this in the case of function approximation with polynomials in Example: 1), and indeed even more that *separate* discretely labeled data. Characterizing the *implicit regularization* [57, 131, 68, 151, 107, 6, 153] induced by the choice of optimization algorithm is thus important to understand properties of the obtained solutions. For linear models, we have a concrete understanding of the solutions obtained by the most common choices of training loss functions and algorithms:

1. If we minimize the logistic loss using gradient descent on separable training data[3], we will converge eventually (albeit very slowly) to the hard-margin SVM [68, 131].

2. If we minimize the square loss on training data using gradient descent while also using an overparameterized model, we will converge to the minimum-ℓ_2-norm interpolation (Theorem 6.1 from [45]) *provided the initialization is equal to zero.*

Conventional wisdom recommends the choice of the logistic loss, or the hinge loss, for classification tasks. For example, the theory of *surrogate losses* [160, 9, 18] gives theoretical arguments favoring the logistic and hinge losses over other convex surrogates, including the square loss. Yet, there have been indications that the reality is more complex, both in underparameterized and overparameterized regimes of ML. For example, [123] extensively compared the hard-margin *support vector machine* (SVM), which minimizes the hinge loss, and *regularized least-squares classification* (RLSC), which minimizes the square loss — ultimately concluding that "the performance of the RLSC is essentially equivalent to that of the SVM across a wide range of problems, and the choice between the two should be based on computational tractability considerations." Quite similar results[4] for a comparison between the square loss and cross-entropy loss have recently been obtained in [63, 20] for a range of modern neural architectures and data sets across several application domains. The latter is the current dominant standard for training neural networks.

A theoretical justification for similar performance of square loss and the cross entropy/logistic loss is provided in [104, 62] and [146] since the SVM itself interpolates one-hot/binary labels under sufficient overparameterization.

[3]The implicit bias has also been characterized for the more difficult non-separable case [68], but we focus here on separable training data as this will always be the case for an overparameterized setting.

[4]In fact, while the results were generally close, in a majority of classification tasks models trained using the square loss outperformed models trained with cross-entropy.

High dimensional linear regression

To understand the double-descent phenomenon observed by [53, 14] better several works study the simpler setting of overparameterized linear regression, fitting a linear model to noisy data when the number of features exceeds the number of training points, for a variety of feature families. The minimum-ℓ_2 norm interpolator is of particular interest and has been studied extensively. (An incomplete list is [59, 97, 10, 15, 105, 19, 75, 152, 121]). To generalize well, the feature family must satisfy a delicate balance between having a few important directions that favor the true signal (unknown function), and a large number of unimportant directions that absorb the noise in a harmless manner. In our work that studies overparameterized linear models in Chapter 2 and 3, we highlight this by use of a bi-level covariance model where the first few features correspond to larger eigenvalues as compared to the rest of the features and such features are favored when performing minimum-ℓ_2-norm interpolation. If the true signal is among these favored features and there are sufficiently many unfavored features then we get both signal preservation and harmless noise absorption.

The minimum-ℓ_1-norm interpolator has also been studied in [105, 99, 87, 145] and while sparsity seeking behavior helps preserve the true signal it poses a challenge for the harmless absorption of noise since the averaging behaviour is not achieved fully.

High dimensional binary classification/logistic regression

Both concurrently with and subsequent to the wave of analyses on overparameterized regression, researchers turned their attention to binary classification. A line of work poses the overparameterized binary classification problem as an optimization problem and analyzes it directly to obtain precise asymptotic behaviours of the generalization error [40, 125, 69, 136, 101, 73, 135]. The key technical tool employed in these works is the Convex Gaussian Min-max Theorem and the resultant error formulas involve solutions to a system of non-linear equations that typically do not admit closed-form expressions. The generalization error of the max-margin SVM has also been analyzed directly by studying the iterates of gradient descent in [25] and leveraging the implicit regularization perspective of optimization algorithms.

However, although the above works did significantly enhance our understanding of binary classification in the overparameterized regime, a fundamental question was not answered: "Is classification easier than regression?" While the classification task is easier than the regression task at test time (regression requires us to correctly predict a real value while binary classification requires us to only predict its sign correctly), the training data for classification is less informative than that for regression since the labels are also binary. Chapter 3 answers this question by exhibiting an asymptotic regime where binary classification error goes to zero, but the regression error does not by considering a bi-level covariance model with Gaussian or Fourier features. It turns out that the level of anisotropy (favoring of true features) required to perform regression correctly is significantly higher than that required for binary classification.

High dimensional multiclass classification

There is a large classical body of work on multiclass classification algorithms [148, 21, 41, 37, 82], with further works giving computationally efficient algorithms for extreme multiclass problems with a huge number of classes [32, 156, 120]. Numerous theoretical works investigate the consistency of classifiers [159, 117, 116, 138, 27]. Finite-sample analysis of the generalization error in multiclass classification problems in the underparameterized regime has been studied in [76, 56, 3, 85, 34, 83, 93, 84, 78, 79] and includes both data dependent bounds using Rademacher complexity, Gaussian complexity and covering numbers as well as data-independent bounds using the VC dimension. Recent work [139] leverages the Convex Gaussian Min-max Theorem to precisely characterize the asymptotic behaviour of the least-squares classifier in underparameterized multiclass classification.

So, how different is multiclass classification from binary classification? The test time task is more difficult and for the same total number of training points, we have fewer positive training examples from each class. We show in our work in Chapter 4 that this poses a challenge while recovering the true signal and consequently makes the multiclass classification task more difficult as compared to the binary classification task.

Several empirical studies comparing the performances of multiclass classification via learning multiple binary classifiers have been undertaken [122, 51, 3]. The effects of the loss function while using deep nets to perform classification has also been investigated [61, 52, 77, 20, 39, 74, 63].

More recently, [146] makes progress towards bridging the gap between empirical observations and theoretical understanding by proving that in certain overparameterized regimes the solution to a multiclass SVM problem is identical to the one obtained by minimum-norm interpolation of one-hot encoded labels (equivalently, that gradient descent on squared loss leads to the same solution as gradient descent on cross-entropy loss as a result of implicit bias of these algorithms [45, 68, 131]). In addition, [146] extends the analysis presented in [104] for the binary classification problem to the multiclass problem with finitely many classes via an interesting reduction to analyzing a finite set of pairwise competitions, all of which must be won for multiclass classification to succeed.

1.5 Related work on learning control strategies

Neural networks for control and communication

Neural networks have been widely used in the past for system identification as well as to learn good control strategies [12, 111]. There has been significant investigation into the use of modular networks for learning to control dynamical systems [65, 64]. More recently in [71], recurrent neural network based architectures have been used for learning feedback codes in communication system leveraging the noisy feedback from the system. The neural network based feedback codes outperform the best hand-crafted schemes. These works have shown that structured networks can improve training and the overall performance by imparting the

correct implicit bias. Chapter 5 builds on older results, and our focus is on using intelligently chosen architectures and training procedures to more easily and robustly learn good neural-network-based control strategies.

The Witsenhausen problem

In 1968, Witsenhausen proposed the Witsenhausen Counterexample [150], where non-linear control strategies could unboundedly outperform linear control strategies [100]. Recent work that used an information-theoretic lens allowed for major insight into the problem and progress towards finding the optimal control strategy (which is still open) [55, 114, 33].

This progress and insight was preceded by computational studies that examined how various strategies performed on the counterexample; in particular, the strategy of "slopey/soft quantization" that played a role in a provably-good strategy in [55] was related to a strategy discovered by Baglietto et al. [8], who used neural networks to find good strategies for the counterexample. The slopey quantization strategy visually resembled scalar instantiations of dirty-paper-coding (DPC) based strategies [35]. DPC approaches give rise to slopey quantization because they involve the quantization of a scaled-down version of the state as an intermediate step.

Further numerical explorations, no matter how they were done [81, 86, 95, 70, 140, 94], always seemed to give rise to strategies that appeared to be close to slopey quantization As a result, the community believes that something smooth that is close to a slopey quantizer is probably optimal or nearly optimal. The challenge is finding it, and more generally, how to find such solutions for more practical decentralized control problems.[5] Can we use neural networks effectively to do this?

In our work in Section 5.2 of Chapter 5 we show that using neural networks for learning control strategies for the Witsenhausen problem is not a straightforward task and choosing an architecture that favors certain structured slopey-quantization-like strategies is required to escape local minimas that exist due to the non-convex nature of the multi-step control problem. In particular, use of a "lattice layer" helps us more easily and more robustly learn good strategies.

Multiplicative noise control system

Simple control systems have been long studied in order to develop an understanding of the fundamental trade-offs involved in communication and control. In our work, we study a system with multiplicative noise with system dynamics given by (1.5). Our problem formulation builds on many previous ideas in information theory and control that have been discussed in depth in books such as [157, 46, 92]. Our specific formulation is inspired by the data-rate theorems [22, 137, 108] as well as the intermittent Kalman Filtering setup [129];

[5] In fact, for the two-controller decentralized infinite-horizon scalar LQG problem, [114, 113] showed that vector counterparts of the Witsenhausen counterexample play a critical role and similar quantization-based strategies are within a constant factor of optimality.

and this formulation was previously discussed in [42, 119]. Additionally, we are inspired by previous works that have studied multiplicative noise including [5, 60, 126, 44, 118]. Xiao et al. [154] and Xu et al. [155] also consider related problems but effectively restrict their attention to LTI strategies, which are not useful in our problem. The results in [42] provide both an upper and lower bound of the largest a for which the system in (1.5) can be stabilized in a second-moment sense. However, the gap between the bounds is enormous, and we believe this is likely due to both of the bounds being loose. It is notable that the optimal linear strategy for this system is $U_n = 0$ for all n, however non-linear strategies can significantly (and unboundedly) improve on the performance of the linear strategy [42]. Here the performance of a control strategy is measured by the largest growth factor a for which it can stabilize the system. A key observation in [42] was that using the controller's memory, i.e. at timestep n using not only the value of Y_n but also the values of Y_{n-1}, Y_{n-2}, etc. in a non-linear way to generate U_n, improved the controller performance.

In Chapter 5, we build on this idea and use neural networks that use the memory of multiple observations to design controllers for this seemingly simple but still-open control problem. We observe that choosing a periodic control structure and a greedy training procedure coupled with input-output scaling across time leads to the correct kind of implicit bias that allows us to learn neural network strategies that perform well.

1.6 Notation

First we describe some basic notation for vectors, matrices, and functions.

Vector and matrix notation

Let $\mathbf{e}_i$ represent the i^{th} standard basis vector (with the dimension implicit). For a given vector $\mathbf{v}$, the functional $\mathsf{sgn}(\mathbf{v})$ denotes the sign operator applied element-wise. Let $\mu_i(\mathbf{M})$ denote the i^{th} largest eigenvalue of positive semidefinite matrix $\mathbf{M}$, and $\mu_{\mathsf{max}}(\mathbf{M})$ and $\mu_{\mathsf{min}}(\mathbf{M})$ denote in particular the maximal and minimal eigenvalue respectively. Further, we use $||\mathbf{M}||_{\mathsf{op}}, \mathsf{tr}(\mathbf{M})$ and $||\mathbf{M}||_{\mathsf{F}}$ to denote the operator norm, trace norm, and Frobenius norm respectively.

Function-specific notation

For two functions $f(n)$ and $g(n)$, we write $f \asymp g$ *iff* there exist universal positive constants (c, C, n_0) such that

$$c\,|g(n)| \leq |f(n)| \leq C\,|g(n)| \;\; \forall n \geq n_0.$$

(In most places where we use the above notation, the functions f and g are positive valued and so we automatically drop the absolute value signs.)

Next, since Chapters 2, 3 and 4 build on top of each other towards one coherent story we summarize the notation used throughout these chapters for the convenience of the reader.

Table 1.1: Notation

Symbol	Definition	Dimension	Source
k	Number of classes	Scalar	
n	Number of training points	Scalar	
d	The total number of features	Scalar	
s	The number of favored features	Scalar	Def. 7
p	Controls overparameterization ($d = n^p$)	Scalar	Def. 7
r	Controls the number of favored features ($s = n^r$)	Scalar	Def. 7
a	Controls the favored weights ($a = n^{-q}$)	Scalar	Def. 7
t	Controls the number of classes ($k = c_k n^t$)	Scalar	Def. 9
c_k	The number of classes when $t = 0$ ($k = c_k n^t$)	Scalar	Def. 9
λ_j	jth eigenvalue of the feature covariance matrix	Scalar	Def. 7
$\mathbf{X}_i$	ith training point's covariate	Abstract	Sec. 2.1
$\phi(\mathbf{X}_i)$	Featurized ith training point's covariate	Length-d vector	Sec. 2.1
Z_i	ith training point's real-valued label	Scalar	Sec. 2.1
Y_i	ith training point's binary-valued label	Scalar	Eqn. 3.2
ℓ_i	ith training point's categorical label	Scalar	Eqn. 4.1
$\boldsymbol{\Sigma}$	Feature covariance matrix	$(d \times d)-$matrix	Sec. 2.1
$\boldsymbol{\alpha}^*$	True coefficients in our well-specified model	Length-d vector	Sec. 2.1
$\boldsymbol{\Phi}_{\mathsf{train}}$	Training feature matrix	$(n \times d)-$matrix	Eqn. 2.1
$\mathbf{Z}_{\mathsf{train}}$	Training real-valued labels	Length-n vector	Sec 2.1
$\mathbf{Y}_{\mathsf{train}}$	Training binary valued labels	Length-n vector	Sec 3.1
$\mathbf{W}_{\mathsf{train}}$	Training additive white Gaussian noise	Length-n vector	Sec 2.1
$\widehat{\boldsymbol{\alpha}}_{\mathsf{ideal}}$	Ideal interpolator that minimizes test error	Length-d vector	Eqn. 2.3
$\widehat{\boldsymbol{\alpha}}_2$	Minimum ℓ_2-norm interpolator	Length-d vector	Eqn. 2.4
$\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}}$	Minimum 2-norm interpolator of binary labels	Length-d vector	Eqn. 3.4
$\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}$	Minimum 2-norm interpolator of real-valued labels	Length-d vector	Eqn. 3.5
$\widehat{\boldsymbol{\alpha}}_{\mathsf{SVM}}$	Hard-margin support vector machine	Length-d vector	Eqn. 3.6
$\mathcal{E}_{\text{reg}}(\widehat{\boldsymbol{\alpha}})$	Regression loss using predictor $\widehat{\boldsymbol{\alpha}}$	Scalar	Def. 1
$\mathcal{E}_{\text{binary}}(\widehat{\boldsymbol{\alpha}})$	Binary classification loss using predictor $\widehat{\boldsymbol{\alpha}}$	Scalar	Def. 5
$\mathcal{E}_{\text{multi}}(\widehat{\boldsymbol{\alpha}})$	Multiclass classification loss using predictor $\widehat{\boldsymbol{\alpha}}$	Scalar	Sec. 4.1

Continued on next page

Table 1.1: Notation (Continued)

Symbol	Definition	Dimension	Source
$\mathsf{SU}(\widehat{\boldsymbol{\alpha}}, \tau)$	Survival for feature τ while using predictor $\widehat{\boldsymbol{\alpha}}$	Scalar	Eqn. 3.20
$\mathsf{CN}(\widehat{\boldsymbol{\alpha}}, \tau)$	Contamination for feature τ while using predictor $\widehat{\boldsymbol{\alpha}}$	Scalar	Eqn. 3.22
$\mathsf{SU}_b(k)$	$\mathsf{SU}_b(k) = \mathsf{SU}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}}, \tau)$	Scalar	Sec. 3.6
$\mathsf{CN}_b(k)$	$\mathsf{CN}_b(k) = \mathsf{CN}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}}, \tau)$	Scalar	Sec. 3.6
$\mathsf{SU}_r(k)$	$\mathsf{SU}_r(k) = \mathsf{SU}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}, \tau)$	Scalar	Sec. 3.6
$\mathsf{CN}_r(k)$	$\mathsf{CN}_r(k) = \mathsf{CN}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}, \tau)$	Scalar	Sec. 3.6
$\mathbf{z}_j$	jth column of the feature matrix $\boldsymbol{\Phi}_{\mathsf{train}}$	Length-n vector	Eqn. 3.26
$\mathbf{A}$	$\mathbf{A} = \boldsymbol{\Phi}_{\mathsf{train}}\boldsymbol{\Phi}_{\mathsf{train}}^\top = \sum_{j=1}^d \lambda_j \mathbf{z}_j \mathbf{z}_j^\top$	$(n \times n)$-matrix	Eqn. 3.26
$\mathbf{A}_{-\tau}$	$\mathbf{A}_{-\tau} = \sum_{j=1, j\neq\tau}^d \lambda_j \mathbf{z}_j \mathbf{z}_j^\top$	$(n \times n)$-matrix	Eqn. 3.26
$\mathbf{Y}^{oh}$	One-hot label matrix	$(n \times k)$-matrix	Eqn. 4.2
$\mathbf{y}_m^{oh}$	One-hot encoding of training points from class m	Length-n vector	Eqn. 4.3
$\mathbf{y}_m$	Zero-mean encoding of training points from class m	Length-n vector	Eqn. 4.4
$\hat{\mathbf{f}}_m$	Learned coefficients use to predict score for class m	Length-d vector	Eqn. 4.7
$\widehat{h}_{\tau,\zeta}$	Relative survival $\widehat{h}_{\tau,\zeta}[j] = \lambda_j^{-1/2}(\hat{f}_\tau[j] - \hat{f}_\zeta[j])$	Length-d vector	Eqn. 4.12
$\mathsf{CN}_{\tau,\zeta}$	Normalizing factor $\mathsf{CN}_{\tau,\zeta} = \sqrt{\left(\sum_{j\notin\{\tau,\zeta\}} \lambda_j^2 (\widehat{h}_{\zeta,\tau}[j])^2\right)}$	Scalar	Eqn. 4.14
$\|\cdot\|_{\psi_2}$	The sub-Gaussian norm of a scalar random variable	Scalar	Eqn. 4.29
$\bar{\mu}$	Center of the eigenvalue bounds for $\mathbf{A}^{-1}$, $\bar{\mu} = \frac{1}{\sum_j \lambda_j}$	Scalar	Eqn. 4.22
$\lozenge$	Deviation term in eigenvalue bounds for $\mathbf{A}$	Scalar	Eqn. 4.32
Δ_μ	Deviation term in eigenvalue bounds for $\mathbf{A}^{-1}$	Scalar	Eqn. 4.24

Chapter 2

Linear regression: A signal-processing perspective

In this chapter, we provide a signal-processing perspective inspired analysis of overparameterized linear regression. We focus primarily on the minimum-ℓ_2-norm interpolator since it admits a closed form expression and moreover, gradient descent on the overparameterized linear regression problem with zero initialization converges to the minimum-ℓ_2-norm interpolator [127, 151]. As mentioned in Section 1.4 of Chapter 1 a couple of excellent papers [59, 10] that center around comprehensive analyses of the ℓ_2-minimizing interpolator were published concurrent to our work [105]. When it comes to whitened or Gaussian features, results that precisely characterize the generalization error of the minimum-norm interpolator are more or less covered across [59] and [10] respectively. The results in these papers use fundamental advances in asymptotic and non-asymptotic random matrix theory. Here, we provide a brief alternative exposition of the main ideas through a Fourier-theoretic lens on *regularly spaced* training data (Example 3). Our aim for doing this is two-fold: one, simply to complement these papers; and two, to provide a signal-processing oriented perspective on salient properties of the minimum-ℓ_2-norm interpolator.

2.1 Problem setup

Throughout, we consider data that is *actually generated* from a well-specified overparameterized linear model[1]. We consider covariate-response pairs $(\mathbf{X}_i, Z_i \in \mathbb{R}^p \times \mathbb{R})_{i=1}^n$ and generative model $Z = \langle \phi(\mathbf{X}), \boldsymbol{\alpha}^* \rangle + W$ for feature vector $\phi(\mathbf{X}) \in \mathbb{R}^d$ and Gaussian *noise* $W \sim \mathcal{N}(0, \sigma^2)$ that is independent of X. We generically assume that the covariates $\{\mathbf{X}_i\}_{i=1}^n$ are iid random samples, but will also consider regularly spaced data on bounded domains for polynomial and Fourier features. The *signal* $\boldsymbol{\alpha}^*$ is unknown apriori to an estimator. We also assume a distribution on $\mathbf{X} \in \mathbb{R}^p$, which induces a distribution on the d-dimensional feature vector $\phi(\mathbf{X})$. Let $\boldsymbol{\Sigma} = \mathbb{E}[\phi(\mathbf{X})\phi(\mathbf{X})^\top]$ denote the covariance matrix of the feature vector under this

[1]The misspecified case is considered in [59, 97].

induced distribution. We assume that $\mathbf{\Sigma}$ is invertible; therefore it is positive definite and its square-root-inverse $\mathbf{\Sigma}^{-1/2}$ exists.

We define shorthand notation for the training data: let

$$\mathbf{\Phi}_{\mathsf{train}} := \begin{bmatrix}\phi(\mathbf{X}_1)^\top & \phi(\mathbf{X}_2)^\top & \dots \phi(\mathbf{X}_n)^\top\end{bmatrix}^\top \tag{2.1}$$

denote the data (feature) matrix, and let $\mathbf{Z}_{\mathsf{train}}, \mathbf{W}_{\mathsf{train}} \in \mathbb{R}^n$ denote the output and noise vectors respectively.

We will primarily consider the overparameterized, or high-dimensional regime, i.e. where $d > n$. We are interested in solutions $\boldsymbol{\alpha}$ that satisfy the following *feasibility condition* for interpolation:

$$\mathbf{\Phi}_{\mathsf{train}}\boldsymbol{\alpha} = \mathbf{Z}_{\mathsf{train}} \tag{2.2}$$

We assume that $\mathrm{rank}(\mathbf{\Phi}_{\mathsf{train}}) = n$, so the set $\{\boldsymbol{\alpha} \in \mathbb{R}^d : \mathbf{\Phi}_{\mathsf{train}}\boldsymbol{\alpha} = \mathbf{Z}_{\mathsf{train}}\}$ is non-empty in $\mathbb{R}^d$.

For any solution $\widehat{\boldsymbol{\alpha}} \in \mathbb{R}^d$, we define the generalization error as test MSE below.

Definition 1. *The expected test mean-squared-error (MSE)* ***minus irreducible noise error*** *of any estimator $\widehat{\boldsymbol{\alpha}}((\mathbf{X}_i, Z_i)_{i=1}^n)$ is given by*

$$\mathcal{E}_{\mathrm{reg}}(\widehat{\boldsymbol{\alpha}}) := \mathbb{E}[(Z - \langle\phi(\mathbf{X}), \widehat{\boldsymbol{\alpha}}\rangle)^2] - \sigma^2,$$

where the expectation is taken ***only*** *over the joint distribution on the fresh test sample $(\mathbf{X}, Z)$, and we subtract off the* ***irreducible*** *noise error $\mathbb{E}[W^2] = \sigma^2$.*

For the well-specified generative model where $Z = \langle\phi(\mathbf{X}), \boldsymbol{\alpha}^*\rangle + W$ the test MSE can be expressed as:

$$\mathcal{E}_{\mathrm{reg}}(\widehat{\boldsymbol{\alpha}}) := \mathbb{E}[\langle\phi(X), \boldsymbol{\alpha}^* - \widehat{\boldsymbol{\alpha}}\rangle^2].$$

We have chosen the convention to subtract off the unavoidable error arising from noise, σ^2, as is standard. From now on, we will denote this quantity to be the test MSE as shorthand.

The training data matrix $\mathbf{\Phi}_{\mathsf{train}}$ can be generated via a number of choices for feature families $\phi(\mathbf{X})$. Some examples are listed below.

Example 2 (Gaussian features)**.** *The Gaussian features on d-dimensional data comprise of $\phi(\mathbf{X}) := \mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{\Sigma} \in \mathbb{R}^{d\times d}$ and $\mathbf{\Sigma} \succ 0$. A special case is iid Gaussian features, i.e. $\mathbf{\Sigma} = \mathbf{I}_d$.*

Example 3 (Fourier features in complex form)**.** *Let $i := \sqrt{-1}$ denote the imaginary number. For one-dimensional data $X \in [0, 1]$, we can write the d-dimensional Fourier features in their* ***complex form*** *as*

$$\phi(X) = \begin{bmatrix}1 & e^{2\pi i X} & e^{2\pi(2i)X} & \dots & e^{2\pi((d-1)i))X}\end{bmatrix} \in \mathbb{C}^d.$$

This is clearly an ***orthonormal*** *feature family in the sense that* $\mathbb{E}_{X\sim Unif[0,1]}\left[\phi(X)_j\phi(X)_k^*\right] = \delta_{j,k}$*, where* $\delta_{j,k}$ *denotes the Kronecker delta and* $(\cdot)^*$ *denotes the complex conjugate.*

We will consider one of two models for the data $\{X_j\}_{j=1}^n$*:*

1. *n-regularly spaced training data points, i.e.* $X_j = \frac{(j-1)}{n}$ *for all* $j \in [n]$*, which we consider empirically* ***and*** *theoretically in Section 2.4.*

2. *n-random training data points, i.e.* X_j *i.i.d* $\sim$ *Unif*$[0,1]$*, which we evaluate only empirically.*

Example 4 (Fourier features in real form)**.** *For one-dimensional data* $X \in [-\pi, \pi]$*, we can write the d-dimensional Fourier features in their* ***real form*** *as*

$$\phi(X) = \begin{bmatrix} \frac{1}{\sqrt{2\pi}} & \frac{1}{\sqrt{\pi}}\sin(x) & \frac{1}{\sqrt{\pi}}\cos(x) & \dots & \frac{1}{\sqrt{\pi}}\sin\left(\frac{d-1}{2}x\right) & \frac{1}{\sqrt{\pi}}\cos\left(\frac{d-1}{2}x\right) \end{bmatrix} \in \mathbb{R}^d,$$

where we assumed that d is odd. This is an orthonormal feature family in the sense that $\mathbb{E}_{X\sim Unif[-\pi,\pi]}\left[\phi(X)_j\phi(X)_k\right] = \delta_{j,k}$ *where* $\delta_{j,k}$ *denotes the Kronecker delta.*

We will consider one of two models for the data $\{X_i\}_{i=1}^n$*:*

1. *n-regularly spaced training data points, i.e.* $X_i = -\pi + \frac{\pi}{n} + \frac{2\pi(i-1)}{n}$ *for all* $i \in [n]$*.*

2. *n-random training data points, i.e.* X_i *i.i.d* $\sim$ *Unif*$[-\pi, \pi]$*.*

Example 5 (Legendre polynomial features)**.** *For one-dimensional data* $X \in [-1, 1]$*, we can write the d-dimensional Vandermonde features as*

$$\phi(X) = \begin{bmatrix} 1 & X & X^2 & \dots X^{d-1}. \end{bmatrix}$$

We can also uniquely define their orthonormalization with respect to the uniform measure on $[-1, 1]$*. In other words, we define the d-dimensional Legendre features as polynomials*

$$\phi(X) = \begin{bmatrix} p_0(X) & p_1(X) & \dots p_{d-1}(X), \end{bmatrix}$$

where $deg(p_j(X)) = j$ *for every* $j \geq 0$*, and* $\{p_j(X)\}_{j\geq 0}$ *are defined such that* $\mathbb{E}_{X\sim Unif[-1,1]}\left[p_j(X)p_k(X)\right] = \delta_{j,k}$*, i.e. the Legendre polynomials form an orthonormal basis with respect to the uniform measure on* $[-1, 1]$*. When evaluating interpolating solutions for both these polynomial features, we consider one of two models for the training data* $\{X_i\}_{i=1}^n$*:*

1. *n-regularly spaced training data points, i.e.* $x_i = -1 + \frac{2(i-1)}{n}$ *for all* $i \in [n]$*.*

2. *n-random training data points, i.e.* X_i *i.i.d* $\sim$ *Unif*$[-1, 1]$*.*

The ideal interpolator $\widehat{\boldsymbol{\alpha}}_{\mathsf{ideal}}$ is one that minimizes the test MSE subject to the interpolating constraint and is defined as:

$$\begin{aligned} \widehat{\boldsymbol{\alpha}}_{\mathsf{ideal}} &= \min_{\boldsymbol{\alpha}} \mathcal{E}_{\mathrm{reg}}(\boldsymbol{\alpha}) \\ \text{s.t. } & \boldsymbol{\Phi}_{\mathsf{train}}\boldsymbol{\alpha} = \mathbf{Z}_{\mathsf{train}}. \end{aligned} \tag{2.3}$$

From [105], we know that the ideal interpolator involves finding the minimum-ℓ_2-norm interpolator of (effectively) *pure noise* using appropriately whitened features. Thus, for overparameterized linear models, it is natural to consider the minimum-ℓ_2-norm solution that interpolates the data as defined below:

Definition 2. *The minimum-ℓ_2-norm interpolator is defined as:*

$$\begin{aligned} \widehat{\boldsymbol{\alpha}}_2 &:= \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^d} \|\boldsymbol{\alpha}\|_2 \\ \text{s.t. } & \boldsymbol{\Phi}_{\mathsf{train}}\boldsymbol{\alpha} = \mathbf{Z}_{\mathsf{train}}. \end{aligned} \tag{2.4}$$

2.2 The minimum-ℓ_2-norm interpolator through the Fourier lens

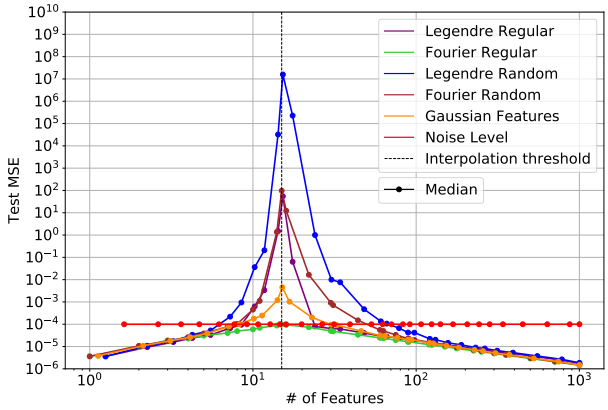
The minimum-ℓ_2-norm interpolator is explicitly characterizable as a linear matrix operator on the output vector $\mathbf{Z}_{\mathsf{train}}$, and can be easily computed as well. When $\boldsymbol{\Phi}_{\mathsf{train}}$ has full row rank (i.e it is of rank n) then we have the closed form expression:

$$\widehat{\boldsymbol{\alpha}}_2 = \boldsymbol{\Phi}_{\mathsf{train}}^\top \left(\boldsymbol{\Phi}_{\mathsf{train}}\boldsymbol{\Phi}_{\mathsf{train}}^\top\right)^{-1} \mathbf{Z}_{\mathsf{train}}.$$

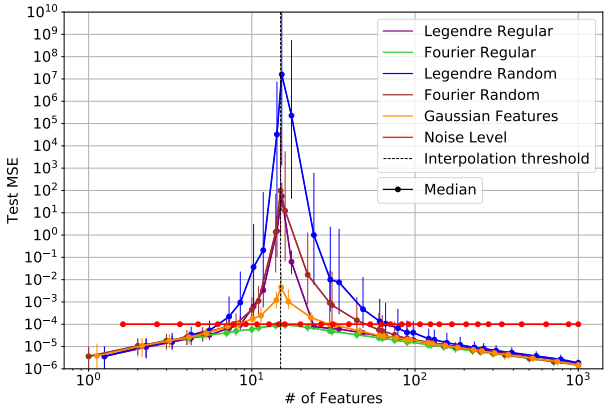
Corollary 1 in [105] shows that for Gaussian features the fundamental price of interpolation of noise on test MSE scales as $\Theta\left(\frac{n}{d}\sigma^2\right)$, where n is the number of training samples, d is the number of features, and σ^2 is the noise variance. Looking at Figure 2.1, we see that when d is large, this is also the scaling that is achieved by the case of regularly spaced data points with Fourier features. This case, as an easy-to-understand paradigmatic example, provides a clear lens into understanding what is happening conceptually *for ℓ_2-minimizing solutions*[2]. It is first useful see how the minimum-ℓ_2-norm interpolator actually behaves for two contrasting examples.

Example 6 (Standard Gaussian features, $k = 500$-sparse signal)**.** *We consider d-dimensional iid standard Gaussian features, i.e. Example 2 with $\boldsymbol{\Sigma} = \mathbf{I}_d$. In other words, the features $\{\phi(\mathbf{X})_j\}_{j=1}^d$ are iid and distributed as $\mathcal{N}(0,1)$. Let the first 500 entries of the true signal $\boldsymbol{\alpha}^*$ be non-zero and the rest be zero, i.e. $\mathsf{supp}(\boldsymbol{\alpha}^*) = [500]$. We take $n = 2000$ measurements, each of which is corrupted by Gaussian noise of variance $\sigma^2 = 0.01$.*

[2]What we will use is the exact presence of exactly *aliased* higher-frequency features corresponding to any low-frequency features. This will give extremely clean expressions for the adverse effect that ℓ_2-minimization has on recovering a signal with low-frequency components.



(a) Median plots for $\mathcal{E}^*_{\text{reg}}$.



(b) Median plots with error bars. Notice the variability near the interpolation threshold.

Figure 2.1. The converse bounds $\mathcal{E}^*_{\text{reg}}$ for interpolation, plotted on a log-log scale for $n = 15$ training points. Here, the median is plotted for clarity where the randomness is due to how the training points are drawn. Notice that all the curves overlap in the significantly overparameterized regime, i.e. $d/n \geq 10$. Here, regular refers to training points chosen on a regularly spaced grid. Random refers to training points chosen uniformly at random. The curves are dithered slightly for readability.Below the interpolation threshold at $d = 15$, the performance of OLS is plotted since interpolation isn't possible there.

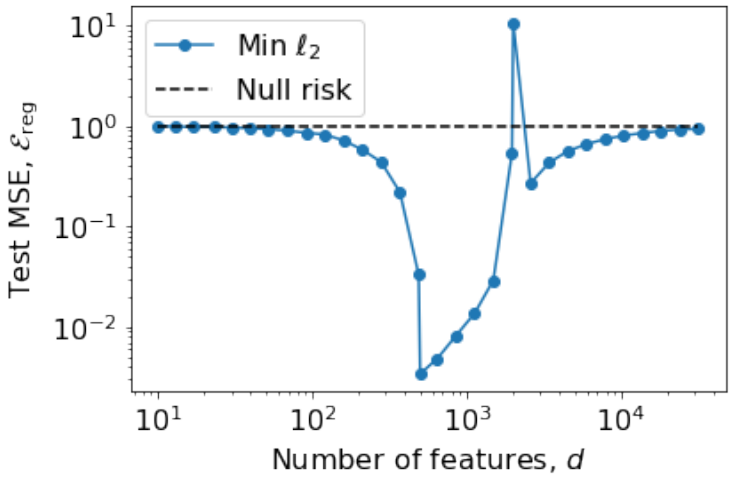


Figure 2.2. Test MSE for Gaussian data sampled from $\mathcal{N}(0, 1)$. Here, $n = 2000$ and $k = 500$ and noise $W \sim \mathcal{N}(0, 0.01)$. Notice the double descent behavior of the test MSE. For $d < n$, we have the U-shaped behavior where in the regime $d < k$, increasing d leads to lower test MSE since we get access to more features present in the true signal. For $k < d < n$, the test MSE increases with a sharp peak at the interpolation threshold ($d = n$). For $d > n$, increasing d initially leads to a sharp decrease in test MSE due to lesser ill-conditioning of the feature matrix but eventually the test MSE rises to the null risk.

Figure 2.2 shows the test MSE of the minimum-ℓ_2-norm interpolator on Example 6 as a function of the number of features d. We immediately notice that the test MSE is converging to the same level as the test error from simply using a hypothetical $\widehat{\boldsymbol{\alpha}} = \mathbf{0}$. This property of generalization error degenerating to that of simply predicting 0 was also pointed out in [59] where this level was called out as the "null risk."

Example 7 (Standard Gaussian features plus a constant, constant signal)**.** *In this example, we consider d-dimensional iid Gaussian features with unit mean and variance equal to* 0.01*. More precisely, the features $\{\phi(\mathbf{X})_j\}_{j=1}^d$ are iid and distributed as $\mathcal{N}(1, 0.01)$. We also assume the generative model for the training data:*

$$Z = 1 + W \tag{2.5}$$

where as before, $W \sim \mathcal{N}(0, \sigma^2)$ is the observation noise in the training data, and we pick $\sigma^2 = 0.01$. Note that in this example the true "signal" is the constant 1*, which is not* ***exactly*** *expressible as a linear combination of the d Gaussian features.*

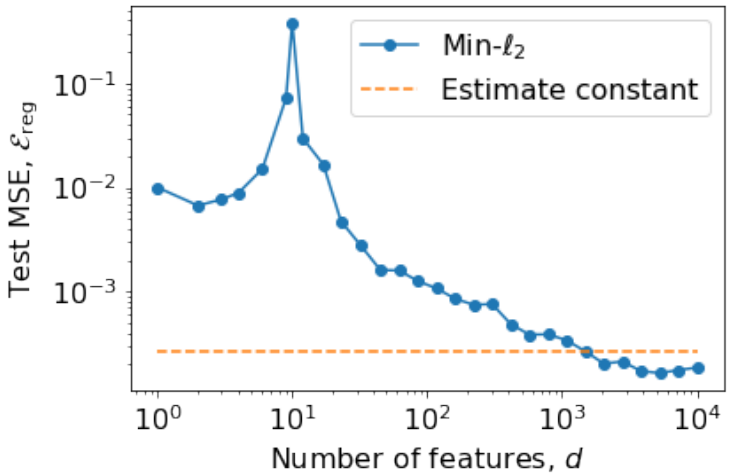


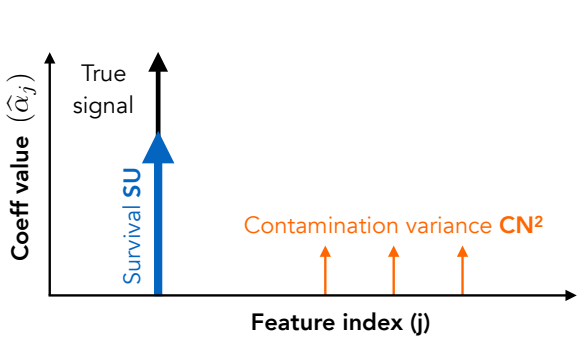
Figure 2.3. Log-log plot for test MSE for the min 2-norm interpolator (ordinary-least-squares to the left of the peak) vs the number of features for i.i.d. Gaussian features $\sim \mathcal{N}(1, 0.01)$. Notice the clear double descent behavior when $d > n$. Further, for sufficiently large d the min 2-norm interpolator has lower test MSE than what we would obtain by simply estimating a constant (i.e mean of the observations). Here $n = 10$ and the true signal is the constant 1. This is Example 7.

Example 7 is directly inspired by feature models[3] in several recent papers [15, 14, 10], as well as having philosophical connections to other recent papers [97, 59]. Figure 2.3 shows

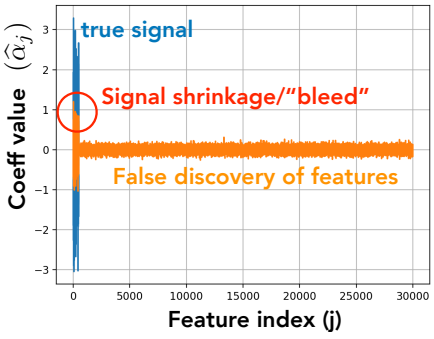
[3] A more general version of this example would replace the constant 1 in the means of the features by the relevant realizations of an underlying latent Gaussian random feature vector with the true signal being that latent Gaussian feature vector. The qualitative behavior of double-descent will be retained, and a formal discussion of this is provided in [104].

the performance of the minimum-ℓ_2-norm interpolator on Example 7 as we increase the number of features. Here, we clearly see the double-descent behavior as the test MSE of the minimum-ℓ_2-norm interpolator decreases with increased overparameterization. Note from Equation (2.5) that the true signal is not exactly representable by a linear combination of the random Gaussian features, and in fact Example 7 is an instance of the linear model being misspecified for any (finite) number of features d. Improved approximability from adding more features to the model partially explains the double descent behavior, but it is not the main reason. It turns out that we would still see the double-descent behavior with the minimum-ℓ_2-norm interpolator if we added another feature that was always the constant 1.

The minimum-ℓ_2-norm interpolator generalizes well for Example 7, showing double descent behavior – but extremely poorly for Example 6. Why does one case fail while the other case works? Bartlett, Long, Lugosi and Tsigler [10] give an account of what is happening directly using the language of random matrix theory. Their paper defines a distinct pair of "effective ranks" using the spectrum of the covariance matrix $\mathbf{\Sigma}$ to state their necessary and sufficient conditions for this interpolator generalizing well. Classic core concepts in signal processing provide an alternative lens to understand these conditions. To use the Fourier-theoretic lens, we will naturally map the number of *regularly spaced* training samples n to what is called the *sampling rate* in signal processing, and the number of features d to what is called the *bandwidth* in signal processing. What we call overparameterization is what is called potential *undersampling* in the signal-processing literature.



(a) Illustration of the "bleed".



(b) Plot of estimated signal components of minimum-ℓ_2-interpolator for iid Gaussian features. Here, $n = 5000$, $d = 30000$ and the true signal $\boldsymbol{\alpha}^*$ has non-zero entries only in the first 500 features.

Figure 1.3. Depiction of signal components "bleeding" out into spurious features as a result of using the minimum-ℓ_2-norm interpolator. The "bleeding" has two effects: lower "survival" of signal in the original true features, and higher "contamination" by spurious features. (repeated from page 5)

Carrying out this story is what we elaborate on in the next section. The key insights are as follows:
(a) The core issue in overparameterization is that of aliasing — there are many different ways to represent the training data using our features. The classical Fourier case with regularly spaced training samples is an extremely clear version of this because the higher-frequency features look literally identical to their lower-frequency counterparts as far as the regularly spaced training points are concerned.
(b) Without some sort of weighting trying to counteract this, standard 2-norm minimization hedges its bets over all the aliases that look alike as far as the training points are concerned. As a result, the signal energy that should have been assigned to the low-frequency features (assuming that the truth is low-frequency) instead "bleeds out" into higher frequency aliases. The more features there are for the same number of training points, the more aliases there are, and consequently, the worse this bleeding will become. This is depicted in Figure 1.3 which was partially introduced in Section 1.3 and is reproduced here for convenience of the reader.
(c) In the limit of severe overparameterization without any counteracting force, the predictions on a randomly chosen test point will usually be something small — since while the aliases were interfering constructively to hit the training points, they are combining non-coherently on the typical test point chosen uniformly at random on the interval in question. This bleeding-to-zero effect is great for absorbing any noise that was in the training data, but it is a disaster as far as signal recovery is concerned.
(d) The only solution for 2-norm minimization is to put a strong enough prior that can block this signal bleeding effect and cause the low frequencies to be *a priori* favored. If the weights favoring the low frequencies are heavy enough relative to their aliases' weights, then a significant fraction of any true signal present at those low frequencies will survive the inference process. Otherwise, it will be severely attenuated. At the same time, the number of low-frequency features that are favored by the weighting must be sublinear in n the number of samples to make sure that the white noise in the training points does largely get attenuated by the "inference filter."
(e) The 2-norm inference process ends up hallucinating nonzero weights on the higher-frequency aliases of the true signal and these end up contaminating the predictions on test points. This can be viewed as a kind of self-interference that is like ISI in communication.
(f) Harmless interpolation with 2-norm interpolator requires these effects to be balanced, and the elementary calculations here naturally recover exact counterparts of the effective rank conditions of Bartlett, et al.[10].
(g) This survival/contamination lens gives us a fresh way of viewing the role of classical ridge regression — this can be interpreted as a kind of overparameterization, but with features that are guaranteed to never contaminate test points. This tells us instantly that ridge regression won't help with poor generalization due to "signal bleed."[4]

[4]See [105] for the analysis of the more generic Tikhonov regularization based on the language of signal bleed and contamination.

Aliasing — the core issue in overparameterized models

We have mapped overparameterization to undersampling of a true signal. The fundamental issue with undersampling of a signal is one of identifiability: infinitely many solutions, each of which correspond to different signal functions, all happen to agree with each other on the n regularly spaced data points. These different signal functions, of course, disagree everywhere else on the function domain, so the true signal function is not well reconstructed by most of them. This results in increased test MSE when such an incorrect function is used for prediction. Such functions that are different, but agree on the sampled points, are commonly called *aliases* of each other in signal-processing language. Exact aliases naturally appear among the features themselves when the features are Fourier, as we see in the below example.

Example 8 (Fourier features with a constant signal)**.** *Denote $i := \sqrt{-1}$ as the imaginary number. Consider the Fourier features as defined in complex form in Example 3 and regularly spaced input on the interval $[0, 1)$, i.e. $x_j = \frac{j-1}{n}$ for all $j \in [n]$.*

Suppose the true signal is equal to 1 everywhere and the sampling model in the absence of noise is

$$Z_j = 1 \text{ for all } j \in [n], \tag{2.6}$$

The estimator has to interpolate this data with some linear combination of Fourier features[5] $f_k(x) = e^{i2\pi kx}$ for $k = 0 \dots d$.

*A trivial signal function that agrees with Equation (2.6) at all the data points is the first (constant) Fourier feature: $f_0(x) = e^{i2\pi(0)x} = 1$. It is, however, not the only one. For example, the complex feature $f_n(x) = e^{i2\pi(n)x}$ will agree with $f_0(x) = 1$ on all the regularly spaced points $\{x_j\}_{j=1}^n$ by the cyclic property of complex Fourier features (i.e. we have $e^{i2\pi(n)\frac{j-1}{n}} = e^{i2\pi(j-1)} = 1 = f_0(x)$). This is similarly true for features $f_{\ell n}(x)$ for all $\ell = 1, 2, \dots, \frac{d}{n} - 1$, and we thus have $\frac{d}{n} - 1$ **exact aliases**[6] of the true signal function $f_0(x)$ on the regularly spaced data points.*

The above property is not unique to the constant function $f_0(x)$: for any true signal function that contains the complex sinusoid of frequency $k^* \in [n]$, i.e. $f_{k^*}(x) = e^{i2\pi(k^*)x}$, the one-complete-cycle signal function $f_{k^*+n} = e^{i2\pi(k^*+n)x}$ again agrees on the n regularly spaced data points, and for this signal function we again have the $\frac{d}{n}$ exact aliases $f_{k^*+\ell n}$ for all $\ell = 1, 2, \dots, \frac{d}{n} - 1$.

The presence of these aliases will naturally affect signal reconstruction. Before discussing this issue, we show an *advantage* in having aliases: they naturally *absorb* the noise that can

[5] Why are we using complex features for our example instead of the real sines and cosines? Just because keeping track of which feature is an alias of which other feature is less notationally heavy for the complex case. The essential behavior would be identical if we just considered sines and cosines.

[6] These aliases are essentially higher frequency sinusoids that look the same as the low frequency one when regularly sampled at the rate n. This is the classic "movie of a fan under a strobelight" visualization where a fan looks like it is stationary instead of moving at a fast speed!

harm generalization. Critically, as defined in Example 3, the Fourier features are orthonormal to each other in complex function space (where the integral that defines the complex inner product is taken with respect to the uniform measure on $[0, 1)$). If we used only the first n Fourier features (i.e. $f_k(x)$ upto $k = n - 1$) to fit an n-dimensional pure noise vector (as described by $\{Z_j = W_j\}_{j=1}^n$), the coefficients of the n-dimensional fit, i.e. $\{\widehat{\alpha}_k\}_{k=1}^n$, would directly correspond to the appropriate discrete Fourier transform (DFT)[7] of the n-dimensional noise vector. By the appropriate[8] Parseval's relation in signal processing, the expected total energy in the feature domain, i.e. $\mathbb{E}\left[\|\widehat{\boldsymbol{\alpha}}\|_2^2\right]$ would be σ^2, and moreover (due to the isotropy of white/independent Gaussian noise), this energy would be equally spread across all n Fourier features in expectation. That is, for every $k \in [n]$ we would have $\mathbb{E}\left[|\widehat{\alpha}_k|^2\right] = \frac{\sigma^2}{n}$.

Now, consider what happens when we include the $\frac{d}{n} - 1$ higher-frequency aliases corresponding to each lower frequency component $k \in [n]$. This gives us d Fourier features in total, and we now consider the minimum-ℓ_2-norm interpolator of noise using all d features. The following is what will happen:

1. In an effort to minimize ℓ_2-norm, the coefficient (absolute) values will be equally *divided*[9] among the $\frac{d}{n}$ aliased features for every realization of the noise samples, i.e. $|\widehat{\alpha}_{k+\ell n}| = |\widehat{\alpha}_k|$ for all $\ell \in \{1, 2, \ldots, \frac{d}{n} - 1\}$ and for all $k \in [n]$.

2. For each $k \in [n]$, the expected *total* contribution from the low frequency feature k and its aliases is now reduced to $\left(\frac{1}{\left(\frac{d}{n}\right)}\right) \cdot \frac{\sigma^2}{n} = \frac{\sigma^2}{d}$. This results in total $\mathbb{E}\left[\|\widehat{\boldsymbol{\alpha}}\|_2^2\right] = \frac{n}{d}\sigma^2$.

For this case, we have zero signal and thus the test MSE for the (whitened) Fourier features is exactly $\|\widehat{\boldsymbol{\alpha}}\|_2^2$. Thus, we have *exactly* recovered the $\Theta\left(\sigma^2 \frac{n}{d}\right)$ scaling for the ideal MSE that was *bounded* in Corollaries 1 and 7 of [105]. The aliases, when used with minimum-ℓ_2-interpolation of noise, are dissipating noise energy, thus directly reducing its potentially harmful effect on generalization in the average[10] case.

[7]The convention for defining the DFT depends on the chosen normalization. The symmetric/unitary DFT can be viewed as choosing the orthonormal basis vectors given by $\frac{1}{\sqrt{n}} e^{2\pi(k)ix}$ evaluated at n regularly spaced points from $[0, 1)$. The classic DFT is defined by a basis with a different scaling — namely $\frac{1}{n}$ instead of $\frac{1}{\sqrt{n}}$. This results in the classic DFT having a factor of $\frac{1}{n}$ in the inverse DFT. We choose the convention for the DFT which normalizes *in the opposite direction*. The basis vectors are just the un-normalized $e^{2\pi(k)ix}$ evaluated at n regularly spaced points from $[0, 1)$. We do not want any scaling factors in the relevant inverse DFT because we want to get the coefficients of the Fourier features.

[8]The reader can verify that the normalization convention we have chosen for the DFT implies $\|\widehat{\boldsymbol{\alpha}}\|_2^2 = \frac{1}{n}\|\mathbf{W}_{\mathsf{train}}\|_2^2$.

[9]The reader who is familiar with wireless communication theory will be able to relate this to the concept of coherent combining gain.

[10]It is also clear that the average case might be very different than the worst case — a phenomenon intimately connected to the fundamental issue of adversarial examples on neural networks that empirically generalize well [110].

Avoiding signal "bleed"

The problem with ℓ_2-minimizing interpolation is that the above effect of absorbing and dissipating training label energy is generic, whether those labels are signal or noise. Whereas being able to absorb and dissipate training harmful noise energy is a good thing, the same exact effect is harmful when there is true signal. Suppose, as in Equation (2.6), that the true signal was a constant (thus the only true frequency component is $k = 0$). A simple calculation shows that the estimated coefficient of the true function would also be attenuated in exactly the same way, and the absolute value of the coefficient corresponding to the constant feature (i.e. $|\widehat{\alpha}_0| = \frac{n}{d}$) decays to 0 as $d \to \infty$. True signal energy, which should ideally be concentrated in the constant feature, is *bleeding* into its aliases in the inference by the ℓ_2-norm-minimizing interpolating solution. This is what we are seeing in the scaling of the test MSE of the ℓ_2-minimizing interpolator for iid Gaussian features (Example 6, shown in Figure 2.2) as well as the convergence of the test MSE of the minimum-ℓ_2-norm interpolator to the "null risk" proved by Hastie, Tibshirani, Rosset and Montanari [59]. An illustration of this bleeding effect is provided in Figure 2.4(a), and the realization of this effect on actually recovered signal components for Example 6 is shown in Figure 2.4(b).

The asymptotic bleeding of signal is a generic issue for whitened features more generally (see [59, Lemma 2]). This may seem to paint a hopeless picture for the ℓ_2-minimizing interpolator even in the absence of noise – how, then, can it ever work? The key is that we can rescale, and mix, the underlying whitened features to give rise to a *transformed feature family*, with respect to which we seek an interpolating solution that minimizes the ℓ_2-norm of the coefficients *corresponding to these transformed features*. The test MSE of such an interpolator will of course be different from the minimum-ℓ_2-norm interpolator using whitened features. The effective difference arises only through the effective rescaling of the whitened features through this transformation: the manifestation of the rescaling can be explicit (the features $\{\phi(\mathbf{X})_k\}_{k=1}^d$ can be visibly scaled by weights $w_k := \sqrt{\lambda_k}$ for some $\lambda_k > 0$) or implicit (the eigenvalues of the covariance matrix[11] $\boldsymbol{\Sigma}$ corresponding to the transformed features $\phi(\mathbf{X})_k$ correspond to the squared weights λ_k).

Consider Example 8 and rescaling $w_k > 0$ for Fourier feature f_k. For a set of coefficients $\boldsymbol{\alpha} \in \mathbb{R}^d$ corresponding to the original whitened features $\{f_k\}_{k=0}^{d-1}$, we denote the corresponding coefficients for the rescaled features by $\boldsymbol{\beta} \in \mathbb{R}^d$. Then, the minimum-$\ell_2$-norm interpolator with respect to the rescaled feature family is as below, for any output $\{Z_j\}_{j=1}^n$:

$$\widehat{\boldsymbol{\beta}} := \arg\min \|\boldsymbol{\beta}\|_2 \text{ subject to}$$
$$\sum_{k=0}^{d-1} \beta_k w_k f_k(x_j) = Z_j \text{ for all } j \in [n].$$

This would recover equivalent coefficients $\widehat{\boldsymbol{\alpha}}$ for the minimum-*weighted*-ℓ_2-norm interpo-

[11] Bartlett, Long, Lugosi and Tsigler [10] present their results through this implicit viewpoint, but their analysis essentially reduces to the explicit viewpoint after a transformation in the underlying geometry.

lator of the data with weight $\frac{1}{w_k}$ corresponding to feature k, as below:

$$\widehat{\boldsymbol{\alpha}} := \arg\min \sum_{k=0}^{d-1} \frac{\alpha_k^2}{w_k^2} \text{ subject to} \tag{2.7}$$

$$\sum_{k=0}^{d-1} \alpha_k f_k(x_j) = Z_j \text{ for all } j \in [n]. \tag{2.8}$$

Now, consider the case of a constant signal without noise, i.e. $Z_j = 1$ for all $j \in [n]$. We already saw that the true signal function, which is $f_0(x)$, satisfies $f(x_j) = 1$ for all $j \in [n]$, as does each of its $\left(\frac{d}{n} - 1\right)$ aliases $\{f_{\ell n}(x)\}$ for $\ell = 1, 2, \ldots, \frac{d}{n} - 1$. Thus, the coefficients of $\widehat{\boldsymbol{\alpha}}$ will be the unique linear combination of the aliases, with coefficients represented by $\{\widehat{\alpha}_{\ell n}\}$, that minimizes the *re-weighted* ℓ_2-norm subject to the sum of such coefficients being exactly equal to 1 (to satisfy the interpolation constraint). The special case of whitened features corresponds to $w_k = 1$ for all $k \in [d]$, and this intuitively results all aliases contributing equally to the recovered signal function. What happens with non-uniform weights: in particular, what happens when w_k decreases as a function of frequency k? Intuitively, the weighted-ℓ_2-norm objective implies that higher-frequency aliases are *penalized more*, and thus a solution would favor smaller coefficients $\widehat{\alpha}_{\ell n}$ for higher integral values of ℓ. In fact, Section 2.4 shows by the principle of matched filtering that the ℓ_2-minimizing coefficients are precisely

$$\widehat{\alpha}_{\ell n} = \frac{w_{\ell n}^2}{V} \text{ where } V := \sum_{\ell=0}^{d/n-1} w_{\ell n}^2 \text{ for all } \ell = 0, 1, \ldots, \frac{d}{n} - 1. \tag{2.9}$$

Since the true *constant* signal is represented by coefficients $\alpha_0^* = 1$ and zero everywhere else, we are particularly interested in the absolute value of $\widehat{\alpha}_0$: how much of the *true signal component* have we preserved? Then, the simple explicit calculation in Section 2.4 shows that this "survival factor" is essentially[12]

$$\mathsf{SU} := \frac{\widehat{\alpha}_0}{\alpha_0^*} = \frac{w_0^2}{\sum_{\ell=0}^{\frac{d}{n}-1} w_{\ell n}^2}. \tag{2.10}$$

The *inverse* of the survival factor SU, after substituting $\lambda_k := w_k^2$, is very closely related to the first "effective rank" condition introduced by Bartlett, Long, Lugosi and Tsigler to

[12]This survival factor can also be understood as the outcome of a competition between two functions. The true signal f_0 that has squared weight w_0^2, and the most attractive orthogonal alias whose squared weight is $\sum_{\ell=1}^{\frac{d}{n}} w_{\ell n}^2$. The minimum 2-norm interpolator will pick a convex combination of the two by minimizing $\frac{\gamma^2}{w_0^2} + \frac{(1-\gamma)^2}{\sum_{\ell=1}^{\frac{d}{n}} w_{\ell n}^2}$ where γ is the survival factor of the true feature. This is minimized by the answer given here for $\gamma = \mathsf{SU}$.

characterize the *bias* of the minimum-ℓ_2-norm interpolator in [10]. Clearly, the survival factor intimately depends on the *relative weights* placed on different frequencies, how many frequencies there are in consideration, and how many perfect aliases there are (the number of aliases is inversely proportional to the number of training samples n). It is illustrative to rewrite the survival factor SU as

$$\mathsf{SU} := \frac{1}{1 + \frac{\sum_{\ell=1}^{\frac{d}{n}-1} w_{\ell n}^2}{w_0^2}}. \tag{2.11}$$

Equation (2.11) is in a form reminiscent of the classic signal-processing "one-pole-filter transfer function". What matters is the relative weight of the favored feature w_0 to the combined weight of its competing aliases. As long as it is relatively high, i.e. $w_0^2 \gg \sum_{\ell=1}^{\frac{d}{n}-1} w_{\ell n}^2$, the true signal will survive. So in particular, if the weights are such that the sum $\sum_{\ell=1}^{\frac{d}{n}-1} w_{\ell n}^2$ converges even as the number of features grows, the true signal will at least partially survive even as $\frac{d}{n} \to \infty$. Meanwhile, if the sum $\sum_{\ell=1}^{\frac{d}{n}-1} w_{\ell n}^2$ diverges **and does so faster than** w_0^2, the signal energy will completely *bleed* out into the aliases (as happens for the whitened case $w_k = 1$ for all k).

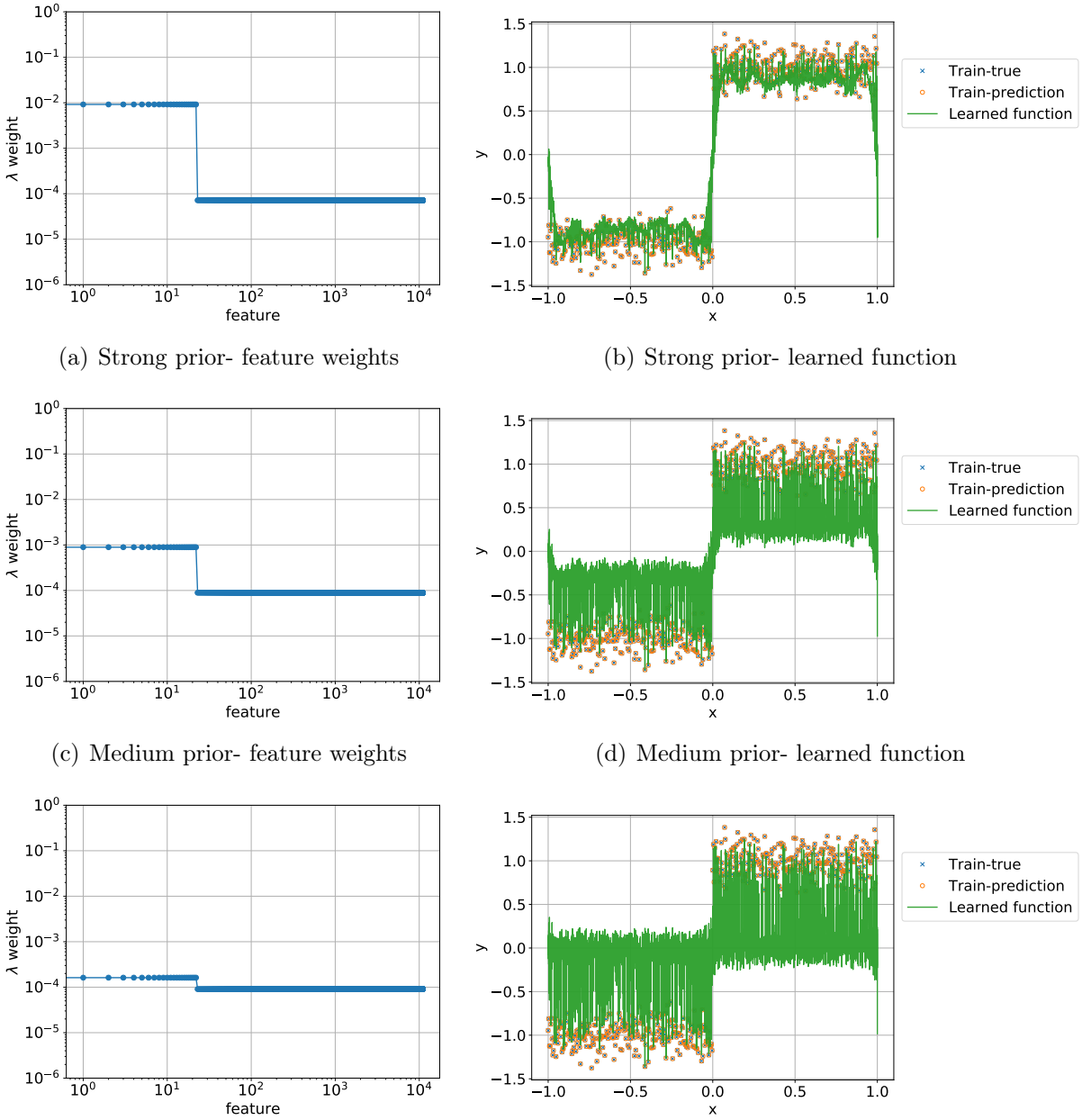
This need for the relative weight on the true features to be high enough relative to their aliases is something that must hold true before any training data has even been collected. In other words, the ability of the 2-norm minimizing interpolator to recover signal is fundamentally restricted. There needs to be a low-dimensional subspace (low frequency signals in our example) that is heavily favored in the weighting, and moreover the true signal needs to be well represented by this subspace. The weights essentially encode an *explicit strong prior*[13] that favors low-frequency features.

We can now start to understand the discrepancy between Examples 6 and 7. There is no prior effect favoring in any way the first 500 features for Example 6. However, by their very nature the features used in Example 7 heavily (implicitly, when the eigenvalue decomposition of $\mathbf{\Sigma}$ is considered[14]) favor the constant feature that best explains the data. This is because the maximal eigenvector of Σ is a "virtual feature" that is an average of the d explicit features, i.e. its entries are iid $\mathcal{N}(1, \frac{0.01}{d})$. This better and better approximates the *constant* feature, the true signal, as d increases – and this improved approximability is the primary explanation for the double descent behavior observed in Figure 2.3.

In Figure 2.4, we illustrate how changing the level of the prior weights impacts interpolative solutions using Fourier features for the simple case of a sign function. Here, there is noise in the training data, but the results would look similar even if there were no training noise — the prior weights are primarily fighting the tendency of the interpolator to bleed signal.

[13]This is in stark contrast to feature selection operators like the Lasso, which select features in a *data-dependent* manner.

[14]In fact, this very case is evaluated in [59, Corollary 1].



(a) Strong prior- feature weights

(b) Strong prior- learned function

(c) Medium prior- feature weights

(d) Medium prior- learned function

(e) Weak prior- feature weights

(f) Weak prior - learned function

Figure 2.4. Effect of different priors on weighted ℓ_2 norm interpolation with $n = 500, d = 11000$ when the true signal is the sign function. The learned function approximates the true signal well when we have a strong prior on the low frequency features but suffers from signal bleed as weaken the prior.

Avoiding signal contamination

We have seen that a sufficiently strong prior in a low-dimensional subspace of features avoids the problem of asymptotically bleeding too much of the signal away — as long as the true signal is largely within that subspace. But what happens when some of the true signal is bled away? How does this impact prediction beyond shrinking the true coefficients? Furthermore, the issue of signal bleed does not by itself answer the question of consistency, particularly with the additional presence of noise. How does the strong prior affect fitting of noise – is it still effectively absorbed by the aliases, as we saw when the features were whitened? To properly understand this, we need to introduce the idea of "signal contamination."

Consider Example 8 now with the *constant-signal-plus-noise* generative model for data:

$$Z_j = 1 + W_j \text{ for all } j \in [n]. \tag{2.12}$$

The output energy (signal as well as noise) bleeds away from the true signal component corresponding to Fourier feature 0 – but because we are exactly interpolating the output data, the energy has to go somewhere. As a result, all energy that is bled from the true feature will go into the aliased features $\{f_{\ell n}\}_{\ell=1}^{d/n-1}$. Each of these features contributes uncorrelated zero-mean unit-variance errors on a test point, scaled by the recovered coefficients $\{\widehat{\alpha}_{\ell n}\}$. Because they are uncorrelated, their variances add and we can thus define the contamination factor

$$\mathsf{CN} := \sqrt{\sum_{\ell=1}^{d/n-1} \widehat{\alpha}_{\ell n}^2}.$$

Even if there were no noise, the test MSE would be at least CN^2. Consequently, it is important to verify that $\mathsf{CN} \to 0$ as $(d, n) \to \infty$.

A straightforward calculation (details in Section 2.4), again through matched-filtering, reveals that the absolute value of the coefficient on aliased feature ℓn is directly proportional to the weight $w_{\ell n}$ and the original true signal strength. Thus contamination (measured as the standard-deviation, rather than the variance in order to have common units), like signal survival, is actually a factor

$$\mathsf{CN} = \frac{\sqrt{\sum_{\ell=1}^{d/n-1} w_{\ell n}^4}}{w_0^2 + \sum_{\ell=1}^{d/n-1} w_{\ell n}^2} \tag{2.13}$$

for the minimum-weighted-ℓ_2-norm interpolator corresponding to weights $\{w_k\}_{k=1}^d$. Substituting $\lambda_k := w_k^2$ results in an error scaling that is very reminiscent of Bartlett, Long, Lugosi and Tsigler's second effective-rank condition. Thus, we see that the two notions of effective ranks[10] correspond to these factors of survival and contamination, which Bartlett, Long, Lugosi and Tsigler sharply characterize for Gaussian features using random matrix theory. The effective "low-frequency features" there represent directions corresponding to the dominant eigenvalues of the covariance matrix $\mathbf{\Sigma}$.

There is a tradeoff: while we saw earlier that the weight distribution described here should somewhat favor low-frequency features (dominant eigenvalue directions), it cannot put too *little weight* on higher-frequency features either. If that happens, the bleeding prevention conditions can be met for a true signal that is in the appropriate low-dimensional subspace. But the noise will give rise to non-vanishing contamination and the variance of the prediction error will not go to 0 as $(n, d) \to \infty$ – then, the minimum-ℓ_2-norm interpolator is inconsistent. Equation (2.13) tells us that the contamination CN will be sufficiently small to ensure consistency *iff* one of the following conditions hold:

1. If the sum of squared alias-weights $\sum_{\ell=1}^{d/n-1} w_{\ell n}^2$ *does not diverge*, the term w_0^2 must dominate this sum in the denominator. Then, we also need $w_0^2 \gg \sqrt{\sum_{\ell=1}^{d/n-1} w_{\ell n}^4}$ so that the denominator dominates the numerator. Note that in this case, the numerator will not diverge either since $\sum_{\ell=1}^{d/n-1} w_{\ell n}^2 \geq \sqrt{\sum_{\ell=1}^{d/n-1} w_{\ell n}^4}$.

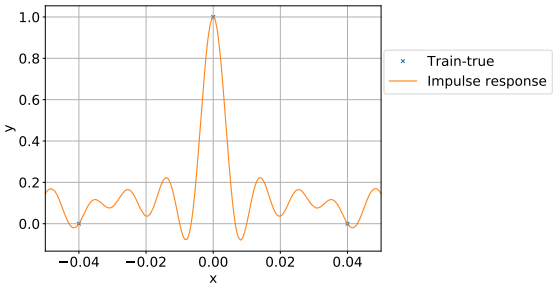
2. The alias-weights $\{w_{\ell n}\}_{\ell \geq 1}$ decay *slowly enough* so that the sum of squared alias-weights $\sum_{\ell=1}^{d/n-1} w_{\ell n}^2$ *diverges*, but decay *fast enough* so that the sum of fourth power of alias-weights $\sum_{\ell=1}^{d/n-1} w_{\ell n}^4$ *does not diverge*. This means that there is sufficient *effective* overparameterization to ensure harmless noise fitting.

3. If the alias-weights decay *slowly enough* that the sum of fourth power of alias-weights *does not diverge* then it must go to infinity at a *slower* rate than the sum of squared alias-weights.

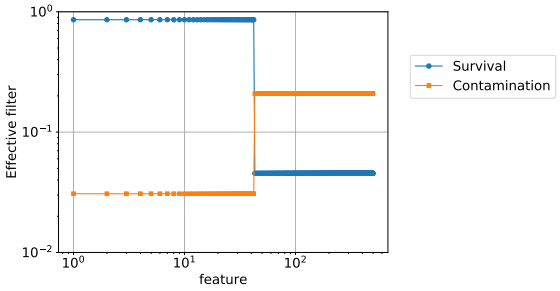
Clearly, avoiding non-zero contamination is its own condition, which is *not* directly implied by avoiding bleeding.

To get consistency, it must be the case that the contamination goes to zero with increasing n, d for everywhere that has true signal as well as an asymptotically complete fraction of the other frequencies. If contamination doesn't go to zero where the signal is, the test predictions will experience a kind of non-vanishing self-interference from the true signal. If it doesn't go to zero for most of where the noise is, then that noise in the training samples will still manifest as variance in predictions.

It is instructive to ask whether the above tradeoff in maximizing signal "survival" and minimizing signal "contamination" manifests as a clean bias-variance tradeoff [14]. The issue is that the contamination can arise through signal and/or noise energy. The fraction of contamination that comes from true signal is mathematically a kind of variance that behaves like traditional bias — it is an estimation error that the inference algorithm makes even when there is no noise. The fraction of contamination that comes from noise is indeed a kind of variance that behaves like traditional variance — it would disappear if there were no noise in training data.



(a) Pulse shaping kernel for $n = 50, d = 354$.



(b) Flter view on the "survival" and contamination for $n = 500, d = 11000$.

Figure 2.5. Weighted ℓ_2 norm interpolation for regularly spaced Fourier features with a strong prior on low frequency features.

A filtering perspective on interpolation

Returning to the case of Fourier features with regularly spaced training points, we can see that given the weightings w_k on all the features, we can break the features into cohorts of perfect aliases. All the features are orthogonal (vis-a-vis the test distribution) and because of the regular sampling, each cohort is orthogonal to every other cohort even when restricted to the n sample points. Consequently, we can understand the bleeding within each of the cohorts separately. Moreover, if we assume that the true signal is going to be low-frequency[15], then we can think about how much the lowest frequency representative of each cohort bleeds. This can be expressed in terms of the survival $0 \leq \mathsf{SU}(k) \leq 1$ for that low-frequency feature k when using the weighted minimum 2-norm interpolator. These $\{\mathsf{SU}(k)\}_{k=0}^{n-1}$ together can be viewed as a filter. This filter tells us how much the act of sampling and estimating attenuates each frequency in the true signal. This attenuation is clearly a kind of "shrinkage."

With the filtering perspective, we can immediately see that for the minimum-weighted-ℓ_2-norm interpolator to succeed in recovering the signal, the true signal needs to be well approximated by the low-frequency features $\{k\}$ for which $\mathsf{SU}(k) \cong 1$ – otherwise the true pattern will be substantially bled away. We also see that to be able to substantially absorb/dissipate the noise energy (which is going to be spread equally across these n cohorts by the isotropic property of white Gaussian noise), it must be the case that most of the survival coefficients $\{\mathsf{SU}(k)\}_{k=0}^{n-1}$ are quite small — most of the noise energy needs to be bled away. As we tend (n, d) to infinity, we can quantify the required conditions for consistency. In the "continuous time" setting, as n is increasing, the continuous-time frequency (that corresponds to the "fastest" feature) is growing with n. So, as long as the maximal value of this frequency k for which the signal would survive (i.e. $\mathsf{SU}(k) \cong 1$) grows *sub-linearly*[16]

[15]This is just for simplicity of exposition and matching the standard machine learning default assumption that all things being equal, we prefer a smoother function to a less smooth function. If the weighting were different, then we could just as well redo this story looking at the highest-weight member of the alias cohort.

[16]If this is reminiscent of the conditions discussed when one considers Nadaraya-Watson kernel estimation in non-parametric statistics, this is no coincidence as [17] points out clearly.

in n, **and** the set of frequencies for which the signal would "bleed out" (i.e. $\mathsf{SU}(k) \to 0$) is asymptotically $n - o(n)$ frequencies, there is hope of both recovering a low-frequency signal as well as absorbing noise.

On one hand, if $\Omega(n)$ of the $\mathsf{SU}(k)$s stay bounded above 0, then those dimensions of the white noise will clearly not be attenuated as desired, and will show up in our test predictions as a classical kind of prediction variance that is not going to zero. On the other hand, if the true signal is not eventually expressible by low-frequency features whose "survival" coefficients approach 1, then there is asymptotically non-zero bias in the prediction.

A further nice aspect of the filtering perspective is that it also lets us immediately see that since the relevant Moore-Penrose pseudo-inverse is a linear operator, we can also view it in "time domain." In machine learning parlance, we could call this the "kernel trick", by which the prediction rule has a direct (and in this case linear) dependence on the labels for the training points. In a traditional signal processing, or wireless communications, perspective, this arises from pulse-shaping filters, or interpolating kernels. A particular set of weights induces both a "survival" filter and an explicit time-domain interpolation function. This is illustrated in Figure 2.5 for a situation in which we put a substantial prior weight on the low-frequency features. Notice that the low-frequency features survive, and have very little contamination. Meanwhile, the higher-frequencies are attenuated, and though their energy is divided across even higher frequency aliases, the net contamination is also small. The time-domain interpolating kernel looks almost like a classical low-pass-filter, except that it passes through zero at the training point intervals to maintain strict interpolation.

2.3 Discussion

In this chapter, via a signal-processing inspired perspective, we identified the key challenges in overparameterized linear regression as "signal bleed" due to shrinkage of true signal and "contamination" due to false discovery of aliases of the true signal and the additive training noise.

For the minimum-ℓ_2-norm interpolator, in the presence of a prior or weighting on the features, the true signal can be preserved while ensuring the noise is dissipated but these two effects can be at odds with each other and thus the prior needs to be appropriately balanced. If we don't have a sufficiently strong prior on the directions (features) where the true signal is present then there is too much "signal bleed". However, if we have too strong of a prior on a few features then the additive training noise cannot be absorbed harmlessly and contaminates our prediction.[17]

A very interesting phenomenon can be observed from Figure 2.4 while using a medium prior. Here, the prior is not sufficiently high to recover the true signal but is high enough that the recovered part of the true signal is able to overcome the contamination. As a

[17]This phenomenon persists beyond the minimum-ℓ_2-norm interpolator and it has been shown that sparsity seeking methods such as the minimum- ℓ_1-norm interpolator recover true signal but struggle to dissipate training noise [105].

consequence, the learned function has the same sign as the true function and can accurately predict where the true function is positive or negative. This prompts the question, are there regimes where we can generalize well (achieve low test error) for binary classification even though we can't generalize well for the regression task? The next chapter shows that this is indeed the case.

2.4 Appendix: Calculations for the regularly spaced Fourier features in complex form

Consider the setup as in Example 3. We have the observation $\mathbf{Z}_{\mathsf{train}} \in \mathbb{C}^n$ and d Fourier features, $f_k(X_{\mathsf{train,j}}) = e^{i2\pi k X_{\mathsf{train,j}}}$ for $k = \{0, 1, \ldots, d-1\}$. Here $\mathbf{X}_{\mathsf{train}}$ contains regularly spaced samples with $X_{\mathsf{train,j}} = \frac{j}{n}$ for $j \in \{0, 1, \ldots, n-1\}$. For ease of exposition we assume $d = (M+1)n$ for positive integer M. We wish to understand how the minimum-weighted ℓ_2-norm interpolating solution behaves in this setting. The first n features, form an orthogonal basis for $\mathbb{C}^n$ and thus it suffices to understand the case where $\mathbf{Z}_{\mathsf{train}}$ is each basis vector separately. Let

$$\begin{aligned} \mathbf{Z}_{\mathsf{train}} &= f_\tau(\mathbf{X}_{\mathsf{train}}), \\ \text{i.e } Z_{\mathsf{train,j}} &= e^{i\pi\tau X_{\mathsf{train,j}}}, j \in [n]. \end{aligned} \tag{2.14}$$

for some $\tau \in \{0, 1, \ldots, n-1\}$.Without loss of generality we consider τ in the range $[0, n-1]$ since subsequent blocks of n features will be aliases of these features.

Note that we can write,

$$\mathbf{Z}_{\mathsf{train}} = \sum_{k=0}^{d-1} \alpha_k^* f_k(\mathbf{X}_{\mathsf{train}}),$$

where $\boldsymbol{\alpha}^* = \mathbf{e}_\tau \in \mathbb{R}^n$ and $\mathbf{e}_k$ denotes the k^{th} standard basis vector. For any solution $\boldsymbol{\alpha} \in \mathbb{C}^n$, the interpolating constraint is,

$$\mathbf{Z}_{\mathsf{train}} = \sum_{k=0}^{d-1} \alpha_k f_k(\mathbf{X}_{\mathsf{train}}).$$

If we scale feature $f_j(\mathbf{X}_{\mathsf{train}})$ by real weight w_j, then the interpolating constraint becomes,

$$\mathbf{Z}_{\mathsf{train}} = \sum_{k=0}^{d-1} \beta_k w_k f_k(\mathbf{X}_{\mathsf{train}}),$$

with $\beta_k = \frac{\alpha_k}{w_k}$.

We are interested in the minimum weighted ℓ_2-norm solution subject to the interpolating constraint given by,

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}\in\mathbb{C}^d} \|\boldsymbol{\Gamma}^{-1}\boldsymbol{\alpha}\|_2 \quad \text{s.t. } \mathbf{Z}_{\mathsf{train}} = \sum_{k=0}^{d-1} \alpha_k f_k(\mathbf{X}_{\mathsf{train}}). \tag{2.15}$$

where $\boldsymbol{\Gamma} = \mathrm{diag}(w_0, w_1, \ldots, w_{d-1})$. Note that this is equivalent to the minimum ℓ_2-minimizing coefficients corresponding to the weighted features, as defined in [10]:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_2 \quad \text{s.t. } \mathbf{Z}_{\mathsf{train}} = \sum_{k=0}^{d-1} \beta_k w_k f_k(\mathbf{X}_{\mathsf{train}}). \tag{2.16}$$

We will solve the problem in (2.16) next. First, we list some properties of the regularly spaced Fourier features. Denote by $S(\tau)$, the set of indices corresponding to features that are exact aliases of $f_\tau(\mathbf{X}_{\mathsf{train}})$. Then,

$$S(\tau) = \{\tau + n, \tau + 2n, \ldots, \tau + Mn\}. \tag{2.17}$$

Note that $M = |S(\tau)| = \frac{d}{n} - 1$. We have,

$$f_k(\mathbf{X}_{\mathsf{train}}) = f_\tau(\mathbf{X}_{\mathsf{train}}),\ k \in S \tag{2.18}$$

$$\langle f_k(\mathbf{X}_{\mathsf{train}}),\ f_\tau(\mathbf{X}_{\mathsf{train}})\rangle = 0,\ k \notin \{\{\tau\} \cup S(\tau)\}. \tag{2.19}$$

Using (2.14), (2.18) and (2.19) we can rewrite the optimization problem in (2.16) as,

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_2 \quad \text{s.t. } \beta_\tau w_\tau + \sum_{k\in S(\tau)} \beta_k w_k = 1.$$

Clearly to minimize the objective we must have $\hat{\beta}_k = 0$ for $k \notin \{\{\tau\} \cup S(\tau)\}$ and thus it suffices to consider the problem restricted to the indices in $\{\{\tau\} \cup S(\tau)\}$. By mapping these indices to the set $\{0, 1, \ldots, M\}$ and denoting the weight vector restricted to this set as $\tilde{\mathbf{w}}$ we write an equivalent optimization problem,

$$\hat{\boldsymbol{\xi}} = \arg\min_{\boldsymbol{\xi}} \|\boldsymbol{\xi}\|_2 \quad \text{s.t. } \sum_{k=0}^{M} \xi_k \tilde{w}_k = 1.$$

Note that $\tilde{w}_k = w_{\tau+kn}$ for $k = 0, 1, \ldots, M$.
We find an optimal solution to this problem by using the Cauchy Schwarz inequality which states,

$$\|\hat{\boldsymbol{\xi}}\|_2 \|\tilde{\mathbf{w}}\| \geq \left| \langle \hat{\boldsymbol{\xi}}, \tilde{\mathbf{w}} \rangle \right|,$$

where equality occurs if and only if $\hat{\boldsymbol{\xi}} = c\tilde{\mathbf{w}}$ for some $c \in \mathbb{C}$. Using the fact that $w_k \in \mathbb{R}$ and solving for c using the interpolating constraint we get,

$$\hat{\boldsymbol{\xi}} = \frac{\tilde{\mathbf{w}}}{\|\tilde{\mathbf{w}}\|_2^2}.$$

Mapping this back to original indices we get,

$$\hat{\beta}_k = \begin{cases} \frac{w_k}{V}, & k \in \{\{\tau\} \cup S(\tau)\} \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\hat{\alpha}_k = \begin{cases} \frac{w_k^2}{V}, & k \in \{\{\tau\} \cup S(\tau)\} \\ 0, & \text{otherwise,} \end{cases} \tag{2.20}$$

where,

$$V = \sum_{k \in \{\{\tau\} \cup S(\tau)\}} w_k^2.$$

Next we consider the effect on a test point $X \in \mathbb{R}$ with i.i.d. entries $X \sim U[0, 1]$, when the ground truth observation is $Z = f_\tau(X)$. On this point we predict,

$$\hat{Z} = \sum_{k=0}^{d-1} \hat{\alpha}_k f_k(X).$$

We want to understand how different $\hat{Z}$ is from Z. Using (2.20) we have,

$$\begin{aligned} \hat{Z} &= \hat{\alpha}_\tau f_\tau(X) + \sum_{k \in S} \hat{\alpha}_k f_k(X) \\ &= \hat{\alpha}_\tau Z + \sum_{k \in S} \hat{\alpha}_k f_k(X). \end{aligned}$$

The prediction $\hat{Z}$ consists of two components. The first component is the true signal attenuated by a factor $\hat{\alpha}_\tau$ due to the effect of signal bleed. The signal bleeds to features that are orthogonal to the true signal and this leads to the second component, a contamination term that we denote by,

$$B = \sum_{k \in S(\tau)} \hat{\alpha}_k f_k(X).$$

Let $\mathsf{SU}(\tau)$ denote the fraction of the true coefficient that survives post signal bleed. Then,

$$\mathsf{SU}(\tau) = \frac{\hat{\alpha}_\tau}{\alpha^*_\tau} = \frac{w_\tau^2}{\sum\limits_{k\in\{\{\tau\}\cup S(\tau)\}} w_k^2}. \tag{2.21}$$

Let $\mathsf{CN}(\tau)$ denote the standard deviation of the contamination given by,

$$\mathsf{CN}(\tau) = \sqrt{\mathbb{E}[|B)|^2]}.$$

Using the property of Fourier features when $\mathbf{X}$ is spaced uniformly in $[0, 1]$ namely,

$$\mathbb{E}[\langle f_i(X),\, f_j(X)\rangle] = \begin{cases} 0, & i \neq j \\ 1, & i = j. \end{cases}$$

to get,

$$\mathbb{E}(|B|^2) = \sum_{k\in S(\tau)} |\hat{\alpha}_k|^2\, \mathbb{E}[|f_k(X)|^2] = \sum_{k\in S(\tau)} |\hat{\alpha}_k|^2 .$$

Using this we have,

$$\mathsf{CN}(\tau) = \sqrt{\sum_{k\in S(\tau)} |\hat{\alpha}_k|^2} = \frac{\sqrt{\sum\limits_{k\in S(\tau)} w_k^4}}{\sum\limits_{k\in\{\{\tau\}\cup S(\tau)\}} w_k^2}. \tag{2.22}$$

Next we consider examples of weighting schemes for a given n, d pair with large enough $\frac{d}{n}$ when the true signal is at τ. The set of indices containing aliases of the true signal is denoted as $S(\tau)$ as in (2.17).

Example 9.

Uniform weights, $w_k = 1$.

$$\hat{\alpha}_k = \begin{cases} \frac{1}{1+\frac{d}{n}-1} = \frac{n}{d}, & k \in \{\{\tau\} \cup S(\tau)\} \\ 0, & \textit{otherwise.} \end{cases}$$

$$\mathsf{SU}(\tau) = \frac{n}{d}.$$

$$\mathsf{CN}(\tau) = \frac{\sqrt{\frac{d}{n}}}{1+\frac{d}{n}-1} = \sqrt{\frac{n}{d}}.$$

Now if we consider a 1-sparse setting where the true signal is simply one of the features (say feature 1), we see that survival corresponding to the true feature scales as n/d and in overparameterized settings where $d > n$ this will not be close to 1, and thus the resulting test MSE for regression will not be zero. Next, we consider spiked weights model that corresponds to bi-level model from Figure 1.4.

Example 10. *Spiked weights on low frequency features: This selects a fraction of energy to put on the favored set of $s < n$ features. Namely. for $a \in [0,1]$ and $s < n$.*

$$w_k = \begin{cases} \sqrt{\frac{ad}{s}}, & 0 \le k < s \\ \sqrt{\frac{(1-a)d}{d-s}}, & \textit{otherwise.} \end{cases}$$

For $0 \le \tau < s$,

$$\hat{\alpha}_k = \begin{cases} \frac{\frac{ad}{s}}{\frac{ad}{s}+(\frac{d}{n}-1).\frac{(1-a)d}{d-s}} \approx \frac{1}{1+\frac{(1-a)}{na(\frac{1}{s}-\frac{1}{d})}} \approx \frac{1}{1+\frac{s}{n}\left(\frac{1}{a}-1\right)}, & k = \tau \\ \frac{\frac{(1-a)d}{d-s}}{\frac{ad}{s}+\left(\frac{d}{n}-1\right)\frac{(1-a)d}{d-s}} = \frac{1}{\frac{a(d-s)}{s(1-a)}+\frac{d}{n}-1} \approx \frac{1}{\left(\frac{a}{1-a}\right)\frac{d}{s}+\frac{d}{n}}, & k \in S(\tau) \\ 0, & \textit{otherwise.} \end{cases}$$

$$\mathsf{SU}(\tau) \approx \frac{1}{1+\frac{s}{n}\left(\frac{1}{a}-1\right)}.$$

$$\mathsf{CN}(\tau) \approx \frac{\sqrt{\frac{d}{n}}}{\left(\frac{a}{1-a}\right)\frac{d}{s}+\frac{d}{n}} = \sqrt{\frac{n}{d}}.\frac{1}{\left(\frac{a}{1-a}\right)\frac{n}{s}+1} = \frac{s}{\sqrt{nd}}.\frac{1}{\left(\frac{a}{1-a}\right)+\frac{s}{n}}.$$

For $s \le \tau < n$,

$$\hat{\alpha}_k = \begin{cases} \frac{1}{1+\frac{d}{n}-1} = \frac{n}{d}, & k \in \{\{\tau\} \cup S(\tau)\} \\ 0, & \textit{otherwise.} \end{cases}$$

$$\mathsf{SU}(\tau) = \frac{n}{d}.$$

$$\mathsf{CN}(\tau) = \frac{\sqrt{\frac{d}{n}}}{1+\frac{d}{n}-1} = \sqrt{\frac{n}{d}}.$$

For this spike model we observe that if $s/(na) \approx 0$, then the survival is close to 1. Further contamination is close to 0 as long as $s \ll n$ and $n \ll d$ and under these conditions the test MSE for regression will be low. We present a detailed quantitative analysis of the test MSE in the following chapter, Chapter 3, corresponding to the qualitative discussion provided here.

Chapter 3

Regression vs binary classification

This chapter highlights the differences between binary classification and regression, using the overparameterized linear model with Gaussian features. Depending on the extent of "effective overparameterization", the minimum-norm solution can:

- succeed at both regression and binary classification,
- succeed at binary classification and fail at regression, or
- fail at both,

as we show in Theorem 2. The intermediate regime of special interest is the one for which minimum-ℓ_2-norm interpolators generalize poorly in regression tasks, but well in binary classification tasks. Underlying these results is a sharp non-asymptotic analysis of the minimum-ℓ_2-norm interpolator *for the binary classification task*. We conceptually link the techniques introduced in recent analysis of this interpolator for the regression task [10] to the binary classification task, using the signal-processing (Fourier-theoretic) interpretation of the overparameterized regime introduced in the previous chapter. The key difference in the binary classification setting is that the training data only consists of binary labels (as opposed to real-valued labels in the regression setting). However, the binary classification task itself is much easier than the regression task (regression is akin to predicting the correct real value, binary classification is akin to predicting the sign correctly).

3.1 Problem setup

Here, we describe the setup for training and test data, evaluation of binary classification and regression tasks, and choices of featurization (in that order).

Data

Let $\mathcal{X}$ denote the space of input data. For binary classification, our training data are *input data-binary label* pairs $(X_1, Y_1), \ldots, (X_n, Y_n)$ taking values in $\mathcal{X} \times \{-1, +1\}$; for regression, the training data are *input data*-real output pairs $(X_1, Z_1), \ldots, (X_n, Z_n)$ taking values in $\mathcal{X} \times \mathbb{R}$. We assume that there is a feature map $\phi\colon \mathcal{X} \to \mathbb{R}^d$, target linear function parameterized by $\boldsymbol{\alpha}^* \in \mathbb{R}^d$, and label noise parameter $0 \leq \nu^* < 1/2$ such that for every $i \in \{1, 2, \ldots, n\}$, we have

$$Z_i = \langle \phi(X_i), \boldsymbol{\alpha}^* \rangle \text{ and} \tag{3.1}$$

$$Y_i = \begin{cases} \mathsf{sgn}(Z_i) \text{ with probability } (1 - \nu^*) \\ -\mathsf{sgn}(Z_i) \text{ with probability } \nu^*. \end{cases} \tag{3.2}$$

Here, the feature map ϕ is known, but the target parameter $\boldsymbol{\alpha}^*$ (which we refer to as the signal) is unknown. The label noise in Y_i is assumed to be independent of everything else.

Let $\phi(x) = \begin{bmatrix} \phi_1(x) & \ldots & \phi_d(x) \end{bmatrix}^T$ for $x \in \mathcal{X}$, i.e. $\phi_j(x)$ is the value of the j^{th} feature in $\phi(x)$. We will consider the training data covariates $\{X_i\}_{i=1}^n$ to be mutually independent and identically distributed (iid). Let $\boldsymbol{\Sigma} = \mathbb{E}[\phi(X)\phi(X)^\top]$ denote the covariance matrix of the feature vector $\phi(X)$ for X following the same distribution as X_i. We assume $\boldsymbol{\Sigma}$ is invertible, so its square-root-inverse $\boldsymbol{\Sigma}^{-1/2}$ exists.

We define shorthand notation for the training data: let

$$\boldsymbol{\Phi}_{\mathsf{train}} := \begin{bmatrix} \phi(X_1) & \phi(X_2) & \cdots & \phi(X_n) \end{bmatrix}^\top \in \mathbb{R}^{n \times d}$$

denote the data (feature) matrix; $\mathbf{Z}_{\mathsf{train}} := \begin{bmatrix} Z_1 & \ldots & Z_n \end{bmatrix}^\top \in \mathbb{R}^n$ denote the regression output vector; and $\mathbf{Y}_{\mathsf{train}} := \begin{bmatrix} Y_1 & \ldots & Y_n \end{bmatrix}^\top$ denote the classification output vector. Note that if there is no label noise (i.e. $\nu^* = 0$), then we have $\mathbf{Y}_{\mathsf{train}} = \mathsf{sgn}(\mathbf{Z}_{\mathsf{train}})$.

Binary classification, regression, and interpolation

The overparameterized regime constitutes the case in which the dimension (or number) of features is greater than the number of samples, i.e. $d \geq n$. We define the two types of solutions starting with interpolating solutions.

Definition 3. *We consider solutions $\boldsymbol{\alpha}$ that satisfy one of the following feasibility conditions for interpolation:*

$$\boldsymbol{\Phi}_{\mathsf{train}} \boldsymbol{\alpha} = \mathbf{Y}_{\mathsf{train}} \text{ or} \tag{3.3a}$$

$$\boldsymbol{\Phi}_{\mathsf{train}} \boldsymbol{\alpha} = \mathbf{Z}_{\mathsf{train}} \tag{3.3b}$$

In particular, we denote the minimum-ℓ_2-norm interpolation on binary labels as

$$\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}} := \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^d} \|\boldsymbol{\alpha}\|_2 \text{ } \textit{s.t. Equation } (3.3\text{a}) \textit{ holds.} \tag{3.4}$$

Similarly, we denote the minimum-ℓ_2-norm interpolation on real labels as

$$\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}} := \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^d} \|\boldsymbol{\alpha}\|_2 \text{ } \textit{s.t. Equation } (3.3\text{b}) \textit{ holds.} \tag{3.5}$$

Recall from our discussion in Section 1.4 of Chapter 1 that these interpolations arise from minimizing the square loss on training data. If we instead minimized the logistic or hinge loss, we would obtain the hard-margin *support vector machine* (SVM), defined below.

Definition 4. *For linearly separable data, the hard-margin Support Vector Machine (SVM) is $\widehat{\boldsymbol{\alpha}}_{\mathsf{SVM}} \in \mathbb{R}^d$, defined by*

$$\begin{aligned} \widehat{\boldsymbol{\alpha}}_{\mathsf{SVM}} := \operatorname*{argmin}_{\boldsymbol{\alpha}\in\mathbb{R}^d} \quad & \|\boldsymbol{\alpha}\|_2 \\ \textit{s.t.} \quad & Y_i \phi(X_i)^\top \boldsymbol{\alpha} \geq 1 \quad \textit{for all } i = 1, \ldots, n. \end{aligned} \tag{3.6}$$

Note that data is defined to be linearly separable iff the constraints in Equation (3.6) *can be feasibly satisfied by some parameter vector $\boldsymbol{\alpha}$.*

As long as $d \geq n$, there is almost surely a solution that interpolates the binary labels $\{Y_i\}_{i=1}^n$ and it satisfies Equation (3.6) with equality for any continuous distribution on the features. Thus, in the overparameterized regime, the training data is trivially linearly separable. Note, however, that the feasibility constraints do not require the SVM solution to interpolate the binary labels.

The standard metrics for test error in regression and binary classification tasks are, respectively, the mean-square-error (MSE) and binary classification error, defined as follows. In these definitions, we have ignored the irreducible error terms arising from possible additive noise in real outputs and label noise in binary outputs respectively. This reflects the practical goal of all prediction to get the underlying true output right, as opposed to matching noisy measurements of that underlying true output.

Recall from Definition 1 that the excess mean-square-error (MSE) of $\widehat{\boldsymbol{\alpha}} \in \mathbb{R}^d$ is given by,

$$\mathcal{E}_{\text{reg}}(\widehat{\boldsymbol{\alpha}}) := \mathbb{E}[\langle \phi(X),\, \boldsymbol{\alpha}^* - \widehat{\boldsymbol{\alpha}} \rangle^2].$$

Definition 5. *The excess binary classification error of $\widehat{\boldsymbol{\alpha}} \in \mathbb{R}^d$ is given by*

$$\begin{aligned} \mathcal{E}_{\text{binary}}(\widehat{\boldsymbol{\alpha}}) &:= \mathbb{E}\left[\mathbb{I}\left[\mathsf{sgn}(\langle \phi(X),\, \boldsymbol{\alpha}^* \rangle) \neq \mathsf{sgn}(\langle \phi(X),\, \widehat{\boldsymbol{\alpha}} \rangle)\right]\right] \\ &= \mathbb{P}\left[\mathsf{sgn}(\langle \phi(X),\, \boldsymbol{\alpha}^* \rangle) \neq \mathsf{sgn}(\langle \phi(X),\, \widehat{\boldsymbol{\alpha}} \rangle)\right]. \end{aligned} \tag{3.7}$$

Here, all expectations (and ensuing probabilities) are only over the random sample X of test data. As is standard, we will characterize the regression and binary classification test errors with high probability over the randomness in the training data $\{X_i, Y_i\}_{i=1}^n$.

As a final comment, we will typically construct an empirical estimate of both test error metrics from n_{test} test samples of data drawn without any label noise. This is for ease of empirical evaluation.

Featurization

We consider zero-mean Gaussian featurization, i.e. for every $i \in \{1, \ldots, n\}$, we have

$$\phi(X_i) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}). \tag{3.8}$$

We denote the spectrum of the (positive definite) covariance matrix $\mathbf{\Sigma}$ by the vector $\boldsymbol{\lambda} := \begin{bmatrix}\lambda_1 & \ldots & \lambda_d\end{bmatrix}$, where the eigenvalues are sorted in descending order, i.e. we have $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d > 0$.

Throughout, we will consider various *overparameterized ensembles* obtained by scaling the covariance parameter $\mathbf{\Sigma}$ as a function of both the number of training data points, n, and the number of features, d. We theoretically characterize the performance of solutions for binary classification and regression tasks using two representative ensembles, defined below.

Definition 6 (Isotropic ensemble(n, d))**.** *The isotropic ensemble, parameterized by (n, d), considers isotropic Gaussian features, $\mathbf{\Sigma} = \mathbf{I}_d$. For this ensemble, we will fix n and study the evolution of various quantities as a function of $d \geq n$.*

Note that the isotropic ensemble constitutes the "maximal" level of effective overparameterization (as defined in the second effective rank in [10]) for a given choice of (n, d).

Next, we describe the bi-level ensemble illustrated in Figure 1.4 and reproduced here for the convenience of the reader. Recall that in the last chapter we saw a qualitative analysis of regression error under this bi-level model and observed that under certain conditions on the parameters of the bi-level model we get low regression test MSE.

Definition 7 (Bi-level ensemble(n, p, q, r))**.** *The bi-level ensemble is parameterized by (n, p, q, r), where[1] $p > 1, 0 \leq r < 1$ and $0 < q < (p - r)$. Here, parameter p controls the extent of artificial overparameterization), r sets the number of preferred features, and q controls the weights on preferred features and thus effective overparameterization. In particular, this ensemble sets parameters*

$$\begin{aligned} d &:= \lfloor n^p \rfloor \\ s &= \lfloor n^r \rfloor \text{ and} \\ a &= n^{-q}. \end{aligned}$$

[1]We restrict (p, q, r) to this range to ensure that a) the regime is truly overparameterized (choice of p), b) the eigenvalues of the ensuing covariance matrix are always positive and ordered correctly (choice of q), c) the number of "high-energy" directions is sub-linear in n (choice of r).

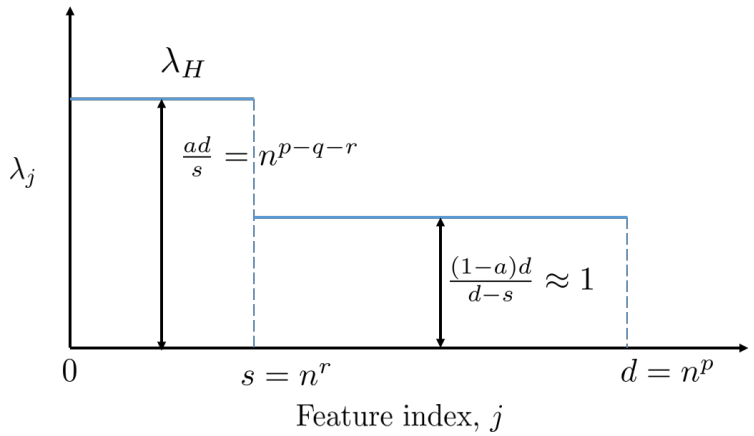


Figure 1.4. The bi-level model parameterized by p, q, r that scales with number of training points n. The number of features $d = n^p$. The covariance matrix has a bi-level structure with the first $s = n^r$ eigenvalues having value n^{p-q-r} while the remaining eigenvalues are approximately 1. (repeated from page 6)

The covariance matrix of the Gaussian features $\mathbf{\Sigma}(p, q, r)$ *is set to be a diagonal matrix, whose entries are given by:*

$$\lambda_j = \begin{cases} \frac{ad}{s}, & 1 \le j \le s \\ \frac{(1-a)d}{d-s}, & \textit{otherwise.} \end{cases}$$

For this ensemble, we will fix (p, q, r) *and study the evolution of various quantities as a function of* n*.*

The bi-level covariance matrix is parameterized by the choice for the top s eigenvalues and the bottom $(d-s)$ eigenvalues, with the sum of eigenvalues being invariant(equal to d). The parameters of critical importance are p, which determines the extent of overparameterization (i.e. number of features), r, which determines the number of larger eigenvalues, and q, which determines the relative values of larger and smaller eigenvalues (all as a function of the number of training points n). We make a few remarks below on this ensemble.

Remark 1. *This bi-level ensemble is inspired by the study of estimation of high-dimensional spiked covariance matrices [e.g. 147, 91] when the number of samples is much smaller than the dimension. In these spiked matrices, the parameter* s *is typically set to a constant (that does not grow with* n*), and the top* s *eigenvalues are highly spiked with respect to the other* $(d-s)$ *eigenvalues. In fact, it is often assumed that there exists a universal positive constant* C*, such that the smaller eigenvalues are bounded and the top (larger) eigenvalues grow with* (d, n) *in the following way:*

$$\lambda_j \ge \frac{d}{Cn} \quad \textit{for all } j \in \{1, \dots, s\} \tag{3.9a}$$

$$\lambda_j \le C \quad \textit{for all } j \in \{s+1, \dots, d\}. \tag{3.9b}$$

Under these conditions, the ratio of the top to the bottom eigenvalues grows as $\Omega\left(\frac{d}{n}\right)$, and Wang and Fan [147] show that the top s estimated eigenvalues of the high-dimensional covariance matrix can be estimated reliably from samples, even when the number of samples is less than the dimension (i.e. $n < d$). This condition, which is also critical for good generalization[2] in regression problems, can be verified to be equivalent to the condition $q \leq (1 - r)$ in our bi-level ensemble (see Theorem 2 for a full statement). Our definition of the bi-level ensemble allows further flexibility in the choice of these parameters, and we will later show that binary classification tasks can generalize well even in the absence of this condition.

Remark 2. *The bi-level ensemble can be verified to match the isotropic ensemble (Definition 6) as a special case when the parameters are set as $q + r = p$. This case represents the maximal level of effective overparameterization, and in general we take $q \leq (p - r)$ to ensure correct ordering of the eigenvalues. The smaller the value of q, the less the effective overparameterization. The models of [25] are spiritually related in how they also use an exponent like q to control the effective overparameterization.*

Remark 3. *We know that for "benign overfitting" [10] of additive noise to occur in regression problems, we need to have sufficiently many (growing super-linearly in n) "unimportant" directions, corresponding to the lower level of eigenvalues. The choice of parameters $p > 1$ and $r < 1$ ensures that the number of such "unimportant" directions is equal to $(d - s) = (n^p - n^r) \gg n$, and so the bi-level ensemble as defined does not admit the regime of harmful overfitting of noise for any choice of parameters (p, q, r). This allows us to isolate signal shrinkage as the principal reason for large regression error, and also study the ramifications of such shrinkage for binary classification error.*

3.2 A Fourier perspective on regression vs binary classification

In Chapter 2, the Fourier features on regularly spaced training data was studied as an "ultra-toy", or caricature model to highlight the consequences of overparameterization in linear regression on noisy data. The ramifications of ℓ_2-minimization are clearly illustrated through this model, as an explicit connection can be made to the classical phenomenon of *aliasing* that is involved to understand the under-sampling of continuous time signals. Using this signal-processing perspective, survival and contamination are natural quantities of interest, as illustrated in Figure 2.4(a) for the 1-sparse case. In Figure 2.4(b), we see how these concepts would *qualitatively* manifest more generally when the underlying signal is hard-sparse.

As we illustrate in this section, appropriate weightings of these features under this "ultra-toy" model also helped us conjecture all of the main results of this chapter. The Fourier

[2] In particular, avoiding signal shrinkage, as also shown in [10].

ensemble is defined below and is an extension of Example 4 where we now have weighted features.

Definition 8 (Weighted Fourier features in real form on regularly spaced data)**.** *We consider n (odd) regularly spaced training points from $(-\pi, +\pi)$ — specifically the sequence $(-\pi + \frac{\pi}{n}, -\pi + \frac{3\pi}{n}, \ldots, -\frac{2\pi}{n}, 0, +\frac{2\pi}{n}, \ldots, +\pi - \frac{\pi}{n})$, a test distribution of X drawn uniformly at random from $(-\pi, +\pi)$, and the d (odd multiple of n) features chosen to be the standard real orthonormal Fourier features:*

$$\frac{1}{\sqrt{2\pi}}, \frac{1}{\sqrt{\pi}}\sin(x), \frac{1}{\sqrt{\pi}}\cos(x), \ldots, \frac{1}{\sqrt{\pi}}\sin\left(\frac{d-1}{2}x\right), \frac{1}{\sqrt{\pi}}\cos\left(\frac{d-1}{2}x\right).$$

For doing minimum-norm interpolation using weighted features as in (2.15), we define the weights corresponding to sines and cosines of frequency j by $\{\lambda_j\}_{j=0}^{\frac{(d-1)}{2}}$. Following the convention of the rest of the chapter, we take the weights $\{\lambda_j\}$ to be a decreasing, strictly positive sequence.

Exact aliases are defined as distinct features that agree with each other (possibly up to a constant multiple) on all the sampled points. The Fourier featurization allows exact aliases to exist. There are three different groups of these exact aliases:

- The initial constant feature is essentially aliased by the cosines at every multiple[3] of n.
- Each cosine feature in the first n features (namely corresponding to a frequency $j \in \{1, 2, \ldots, \frac{n-1}{2}\}$) picks up $(\frac{d}{n} - 1)$ cosine aliases with frequencies $(n - j), (n + j), (2n - j), (2n + j), \ldots$. This is because cosine is an even function and the training samples are symmetrically distributed about 0.
- Similarly, each sine feature in the first n features (corresponding to a frequency $j \in \{1, 2, \ldots, \frac{n-1}{2}\}$) picks up $(\frac{d}{n} - 1)$ sine aliases with frequencies $(n - j), (n + j), (2n - j), (2n+j), \ldots$. However, because sine is an odd function, these aliases have their signs alternating with the $(kn-j)$ ones being multiplied by (-1) and the $(kn+j)$ ones being exact aliases.

Regression vs binary classification

To see a Fourier counterpart of Theorem 2 from Section 3.4, which compares binary classification and regression test error of interpolating solutions, we consider the underlying true function to be $\cos(X)$. At training time, we get actual real-valued outputs $Z_i = \cos(X_i)$ corresponding to the n regularly spaced points $\{X_i\}$.

[3]For ease of exposition, the minor issue of the constant feature having a slightly different scaling vis-a-vis its aliases is going to be ignored in this treatment, but this is simply a matter of keeping track of notation. Alternatively, we could eliminate this by using complex Fourier features. We will finesse this issue here by simply not allowing the true signal to have a constant term in it.

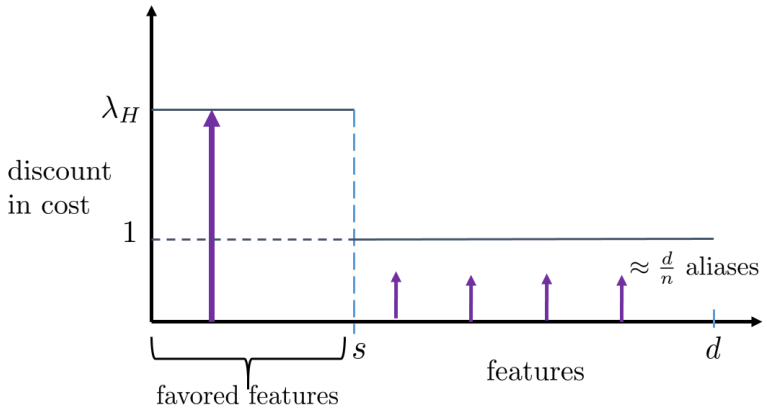


Figure 3.1: An illustration of the bi-level model for the Fourier features.

We consider a bi-level covariance model as in Definition 7 (illustrated in Figure 3.1) where we scale the parameters (s, λ_H, d) with n in a coordinated way. Recall that the number of prioritized features is given by $s := n^r$ for $r \in [0, 1)$, and the number of features $d = n + n^p$ for $p > 1$. (We added an extra term of n to make it easier to count the aliases. This has no asymptotic effect when $p > 1$ and $n \to \infty$.) The λ_H represents how much we favor the special features and in keeping with the scaling in Definition 7, we set $\lambda_H = n^{p-r-q}$ for some $q \in [0, p - r]$.

The minimum-ℓ_2-norm interpolation of real-valued output using the weighted features leads to the following coefficients on the d underlying unweighted Fourier features just as in Section 2.4 from Chapter 2:

$$\widehat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \left(\frac{1}{\lambda_H} \sum_{j=0}^{s-1} \alpha_j^2 \right) + \sum_{j=s}^{d-1} \alpha_j^2$$
$$\text{s.t. } \boldsymbol{\Phi}_{\mathsf{train}} \boldsymbol{\alpha} = \mathbf{z}_{\text{train}}. \tag{3.10}$$

Because of the known orthogonality of the sine and cosine features on n regularly spaced points, the first n columns of $\boldsymbol{\Phi}_{\mathsf{train}}$ are orthogonal. This means that the solution $\widehat{\boldsymbol{\alpha}}$ will only have nonzero entries in the positions that correspond to the $\frac{d}{n} = 1 + n^{p-1}$ different columns of $\boldsymbol{\Phi}_{\mathsf{train}}$ that are copies of the column corresponding to the feature $\cos(x)$. Since $s < n$, exactly one of these will be favored and so the optimization problem in Equation (3.10) turns into the much simpler problem:

$$\min_{a,b \,\big|\, a + n^{p-1} b = 1} \frac{a^2}{n^{p-r-q}} + n^{p-1} b^2 \tag{3.11}$$

where a represents the recovered coefficient corresponding to the true underlying feature $\cos(x)$ and b represents the coefficients on all of its exact aliases.

The calculation in Section 2.4 from Chapter 2 shows that Equation (3.11) is solved by:

$$a = \frac{\lambda_H}{\lambda_H + \left(\frac{d}{n} - 1\right)} = \frac{1}{1 + n^{q-(1-r)}} \text{ and} \tag{3.12a}$$

$$b = \frac{1}{\lambda_H + \left(\frac{d}{n} - 1\right)} = \frac{1}{n^{p-r-q} + n^{p-1}}. \tag{3.12b}$$

Here, a represents the survival of the true signal as in (2.10). For large enough n, this is approximated[4] by

$$a \approx \begin{cases} 1 & \text{if } q < 1 - r \\ n^{-(q-(1-r))} & \text{if } q > 1 - r \end{cases}. \tag{3.13}$$

Equation (3.13) is the Fourier-feature counterpart of the upper and lower bounds on survival in Lemmas 9 (binary labels) and 10 (real-valued output) that we will see in Section 3.7. Now, taking $n \to \infty$, we get

$$a_\infty = \begin{cases} 1 & \text{if } q < (1 - r) \\ 0 & \text{if } q > (1 - r) \end{cases} \tag{3.14}$$

which shows that the signal only fully survives if $q < (1 - r)$.

Let us now measure the *contaminating* effect of falsely discovered features. Following Equation (3.21), we denote $B(X)$ as the random variable that represents the contribution of all of the aliases to the predictions. Each of the Fourier features of non-zero frequency is zero-mean and has variance 1. From the orthonormality (in expectation over test data) of the aliases, we get

$$\begin{aligned} \text{Var}[B(X)] &= n^{p-1} b^2 \\ &= \left(\frac{1}{n^{\frac{p}{2} + \frac{1}{2} - r - q} + n^{\frac{(p-1)}{2}}} \right)^2, \end{aligned} \tag{3.15}$$

where in the last step, we substituted Equation (3.12b). Notice that $\frac{(p-1)}{2} > \frac{p}{2} + \frac{1}{2} - r - q$ whenever $q > (1 - r)$, and so asymptotically we get

$$\mathsf{CN} = \sigma_{CN} \approx \begin{cases} n^{-\left(\frac{p+1}{2} - (q+r)\right)} & \text{if } q < (1 - r) \\ n^{-\frac{(p-1)}{2}} & \text{if } q > (1 - r) \end{cases} \tag{3.16}$$

This expression is the Fourier-feature counterpart of the lower bound on contamination established for Gaussian features in Lemma 13 in Section 3.7.

Thus, provided that $q < (1-r)$, the expression in Equation (3.16) always decays to zero as $n \to \infty$, regardless of which case we are in. The combination of Equations (3.16) and (3.14) tells us that regression in this problem can work to get mean-square-error approaching zero as long as $q < (1 - r)$. On the other hand, when $q > (1 - r)$, signal does not asymptotically survive and regression MSE approaches the null risk.

[4] In the style of the Bode Plot of a one-pole low pass filter.

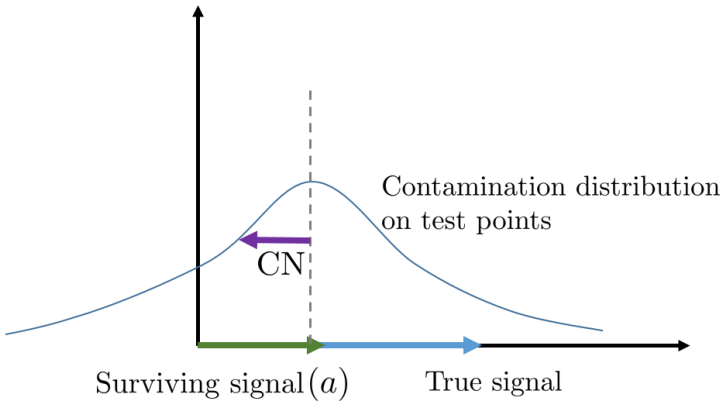


Figure 3.2. Illustration of how contamination can flip the sign of the prediction at a test point. The survival *relative* to the standard deviation of the contamination, CN, is what matters — if the latter is much smaller than the former, then the probability of binary classification error is low.

Implications for binary classification: existence of the separating regime

First, let us assume we have access to real valued outputs $Z_i = \cos(X_i)$ even for the binary classification task. For binary classification, we only care about predicting $\mathsf{sgn}(\cos(X))$ correctly with high probability when $X \sim \text{Unif}[-\pi, \pi]$. Clearly, binary classification also works under the conditions for which regression works (i.e. $q < (1-r)$), but, as we we will see in Theorem 2, can work even in the absence of these conditions. Recall that when $q > (1-r)$, the survival factor $a \to 0$ as $n \to \infty$. However, if the contamination is small enough, i.e. $\sigma_{CN} \ll a$, the probability of binary classification error is extremely low, as illustrated in Figure 3.2. We observe from Equations (3.16) and (3.13) that $\sigma_{CN} \ll a$ if $q < (1-r) + \frac{(p-1)}{2}$. When that happens, binary classification will asymptotically work.

To see this more formally, we can upper bound the expression of binary classification error and show that it goes to zero as $n \to \infty$ under these conditions[5]. We use a union bound together with Chebyshev's inequality in a manner reminiscent of typicality proofs in information theory [36].

Let $\epsilon = \frac{(p-1)}{2} - (q - (1-r))$ be the difference between the relevant two exponents of n corresponding to the ratio a/σ_{CN}. Define the events $\mathcal{A} := \{X \mid \ |\cos(X)| < 2n^{-\frac{\epsilon}{2}}\}$ and $\mathcal{B} := \{X \mid \ |B(X)| > n^{-\frac{\epsilon}{2}} n^{-(q-(1-r))}\}$. The event $\mathcal{A}$ corresponds to having an atypically weak signal in the true feature, and the event $\mathcal{B}$ corresponds to having an atypically large contamination term. Observe that if neither event $\mathcal{A}$ nor event $\mathcal{B}$ holds, we can substitute

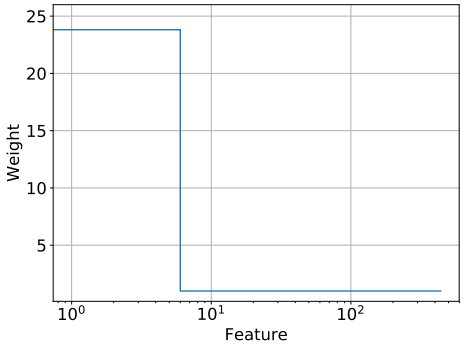
[5]The *exact* Gaussian-feature expression for binary classification error in Proposition 1 depends solely on the ratio a/σ_{CN}. Characterizing the exact expression for Fourier features is more challenging because the contamination does not have a clean distribution, but we can upper bound the probability of binary classification error using the standard deviation alone.

Equation (3.13) to get $|a\cos(X)| \geq 2|B(X)|$, and this implies that a binary classification error will not occur. Therefore, the probability of binary classification error is *upper bounded* by $\mathbb{P}[\mathcal{A} \cup \mathcal{B}]$, and by the union bound it suffices to upper bound the probability of each of these events individually. We start with the "weak signal" event $\mathcal{A}$. Because $\cos(X)$ is a function that is always differentiable in the neighborhood where $\cos(X) = 0$, this means that $\cos(X)$ as a random variable has a density[6] in the neighborhood of 0. Consequently, we have $\mathbb{P}[\mathcal{A}] = \int_{-n^{-\frac{\epsilon}{2}}}^{+n^{-\frac{\epsilon}{2}}} \frac{1}{\pi\sqrt{1-y^2}} dy = \frac{2}{\pi}\sin^{-1}(n^{-\frac{\epsilon}{2}})$ which goes to zero as $n \to \infty$. We now turn to the "unusually large contamination" event $\mathcal{B}$. Because $q < (1-r) + \frac{(p-1)}{2}$, we have $\mathbb{P}[\mathcal{B}] = \mathbb{P}\left[|B(X)| > n^{-\frac{\epsilon}{2}} n^{-(q-(1-r))}\right] \leq \mathbb{P}\left[|B(X)| > n^{\frac{\epsilon}{2}}\sigma_{CN}\right]$. By Chebyshev's inequality, we have $\mathbb{P}[\mathcal{B}] \leq n^{-\epsilon}$, which goes to zero as $n \to \infty$.

Since the probabilities of both events $\mathcal{A}$ and $\mathcal{B}$ have been shown to go to 0 as $n \to \infty$, the limiting binary classification error will also be zero when $q < (1-r) + \frac{(p-1)}{2}$. Finally, it is worth noting that the above calculation only *upper bounds* the binary classification error. Subsequent work [110] show that even when $q > (1-r) + \frac{(p-1)}{2}$ the binary classification error goes to zero because the contamination from the falsely discovered features is concentrated around the training points due to Gibbs phenomenon and is low everywhere else; hence, even though the overall level of contamination may be high, the contribution of the falsely discovered features to prediction on a randomly drawn test point is low.

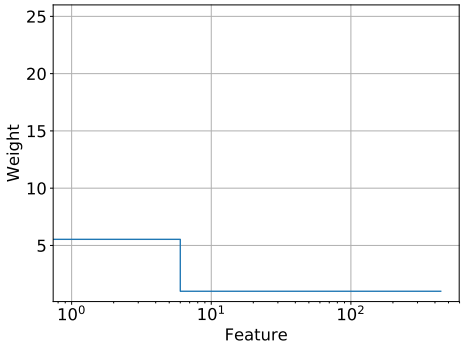
The above argument can be extended to the case of interpolation of *binary labels* by using the Fourier series representation of the underlying true label function. Since there is now misspecification induced by the sign operator, this requires understanding the approximation-theoretic properties of the Fourier series by its first s terms as $s \to \infty$. While analyzing this case theoretically for Fourier features is a challenging task empirical results using Fourier features illustrated in Figure 1.5 introduced in Section 1.3 and reproduced here shows that interpolation of binary labels also admits three regimes including an intermediate regime where binary classification works while regression does not. While the first two regimes display behavior that parallels the Gaussian-feature results in Theorem 2. Note that although for the choice of parameters (in particular a finite value of n) it it appears as if binary classification does not work in the third regime, [110] show that asymptotically binary classification succeeds even in this regime.

[6]This is known as a shifted arc-sine distribution.

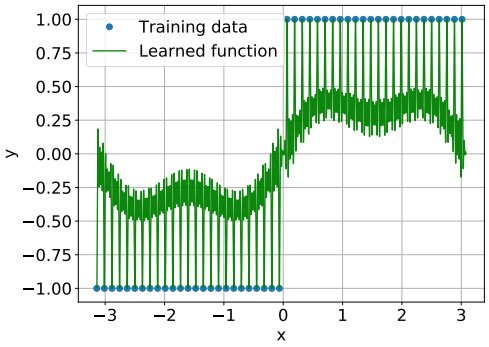


(a) $\lambda_H = 23.81$

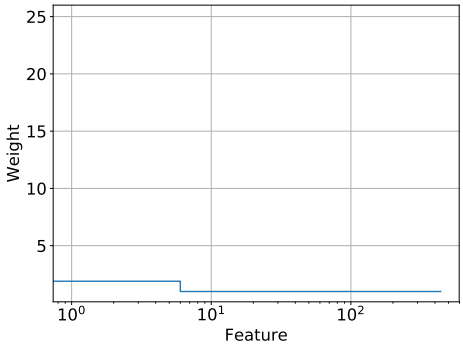
(b) $\lambda_H = 23.81$



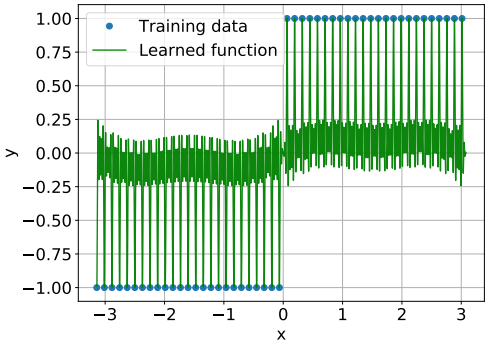
(c) $\lambda_H = 5.53$



(d) $\lambda_H = 5.53$



(e) $\lambda_H = 1.89$



(f) $\lambda_H = 1.89$

Figure 1.5. The three qualitative regimes illustrated using Fourier features and regularly spaced training points. The top corresponds to both regression and binary classification succeeding, the middle one is the intermediate regime where only binary classification works, and the bottom one is where neither works. Here $n = 49, s = 7, d = 441$. (repeated from page 7)

3.3 Comment on binary label interpolation vs SVM

In our work, we study binary classification via minimum-norm interpolation of binary labels (3.4) while the more popular approach for performing binary classification is via the hard-margin SVM (3.6). From the optimization objective and constraints defined in Equation (3.6), we can see that there is a continuum of margins, defined by $Y_i \cdot \phi(X_i)^\top \alpha$, that is possible for each training point. Thus, unlike in least-squares regression, even obtaining an exact expression for the margin-maximizing SVM solution, $\widehat{\boldsymbol{\alpha}}_{\mathsf{SVM}}$, appears difficult in the overparameterized regime. The heart of our approach is to study the minimum-ℓ_2-norm interpolation of binary labels and leverage the result in Theorem 11 from [104] (reproduced below for convenience of reader) that shows that *all the training data points usually become support vectors* in a sufficiently overparameterized regime. In such a setting the solution obtained via binary label interpolation is identical to the one obtained via SVM and thus by analyzing the behavior or the binary label interpolator we can understand what happens when we use the SVM solution instead.

Theorem 1. *(Theorem 11 in [104]) Let $\boldsymbol{\Phi}_{\mathsf{train}}$ follow the Gaussian featurization from Equation* (3.8) *with covariance matrix $\boldsymbol{\Sigma}$, and let $\widehat{\boldsymbol{\alpha}}_{\mathsf{SVM}}$ be the solution to the optimization problem in Equation* (3.6)*.*

1. *If $\boldsymbol{\Sigma}$ satisfies*

$$||\boldsymbol{\lambda}||_1 \geq 72 \left(||\boldsymbol{\lambda}||_2 \cdot n\sqrt{\ln n} + ||\boldsymbol{\lambda}||_\infty \cdot n\sqrt{n} \ln n + 1 \right), \tag{3.17}$$

   the vector $\widehat{\boldsymbol{\alpha}}_{\mathsf{SVM}}$ satisfies the binary label interpolation constraint (Equation (3.3a)*) simultaneously for every $\mathbf{Y}_{\mathsf{train}} \in \{\pm 1\}^n$ with probability at least $\left(1 - \frac{2}{n}\right)$.*

2. *If $\boldsymbol{\Sigma} = \mathbf{I}_d$ (i.e., $\boldsymbol{\Phi}_{\mathsf{train}}$ follows the isotropic ensemble), and*

$$d > 10n \ln n + n - 1, \tag{3.18}$$

   then the vector $\widehat{\boldsymbol{\alpha}}_{\mathsf{SVM}}$ satisfies the binary label interpolation constraint (Equation (3.3a)*) for any fixed $\mathbf{Y}_{\mathsf{train}} \in \{\pm 1\}^n$ with probability at least $\left(1 - \frac{2}{n}\right)$.*

We now remark on this result for the bi-level ensemble. Plugging in the condition from Equation (3.17) into the bi-level ensemble (Definition 7), the following conditions on (p, q, r) are *sufficient* for all training points to become support vectors with high probability (see Appendix C in [104] for a full calculation):

$$p > 2 \text{ and} \tag{3.19a}$$

$$q > \left(\frac{3}{2} - r \right). \tag{3.19b}$$

There is an intuitive interpretation for each of these conditions in light of the second "effective rank" condition that is sufficient for benign overfitting [10] of noise (although our proof

technique is quite different). First, the condition $p > 2$ mandates an excessively large number of unimportant directions, i.e. corresponding to lower-level (smaller) eigenvalues ($(n^p - n^r)$ of them). Second, the condition $q > \left(\frac{3}{2} - r\right)$ mandates that the ratio between the important directions, i.e. higher-level eigenvalues, and the unimportant directions, is sufficiently small — thus, the unimportant directions are sufficiently weighted. This second condition appears to be strictly stronger than what is required for benign overfitting of noise.

Equation (3.19) is quite strong as a sufficient condition, but nevertheless admits non-trivial regimes for which binary classification can generalize well or poorly (see the text accompanying Theorem 2 for a full discussion). Subsequent work to ours [62] tightened the condition in Equation (3.17) by providing a new deterministic equivalent to the phenomenon of all training points becoming support vectors. It suffices to have $q+r > 1$ and $p > 1$ for all training points to become support vectors with high probability. In particular the condition $q + r > 3/2$ was tightened to $q + r > 1$ and the condition $p > 2$ was tightened to $p > 1$.

3.4 Generalization analysis for interpolating solution with Gaussian features

In this section, we attempt an approximate characterization of the ensuing classification error of *minimum-ℓ_2-norm* interpolation on binary labels, denoted by $\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}}$. Our hope is that we can leverage comprehensive analyses of minimum-ℓ_2-norm interpolation for least-squares regression [10, 105]. However, it turns out that direct plug-ins of these analyses do not work for a number of reasons:

1. Even with clean data (i.e. zero label noise), the binary classification setup admits misspecification noise of the form $Y_i - \boldsymbol{\phi}(X_i)^\top \boldsymbol{\alpha}^*$. The misspecification noise is clearly non-zero mean, and is non-trivially correlated with the features. This resists a clean decomposition of generalization error into the error arising from signal identifiability (or lack thereof) + error arising from overfitting of noise, as in [10].

2. For a given interpolation $\widehat{\boldsymbol{\alpha}}$, the expression for binary classification error is distinctly different from mean-square-error (we will see this explicitly in Theorem 1). In particular, we will see that characterizing this expression sharply requires novel analysis of the individual recovered coefficients as a result of interpolation.

Our analysis is subsequently non-trivial to engage with both of these difficulties, and directly addresses both of them by analyzing the minimum-ℓ_2-norm interpolator of binary labels from first principles. This is, roughly speaking, in two steps: first, by characterizing the expected generalization error in terms of 0-1 classification loss *for any solution* (regardless of whether it interpolates or not) as a function of *survival* and *contamination* factors; second, by obtaining sharp characterizations of these factors for the minimum-ℓ_2-norm interpolator of binary labels.

Setup and result

We state our main result for this section in the context of the bi-level ensemble (Definition 7). We fix parameters $p > 1$ (which represents the extent of artificial overparameterization), and $r \in [0, 1)$ (which sets the number of preferred features), and $q \in [0, p - r]$ (which controls the weights on preferred features, thus effective overparameterization); and study the evolution of regression and binary classification risk as a function of n. For the purpose of this section, we denote the regression and binary classification test losses under the bi-level ensemble as $\mathcal{E}_{\text{reg}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}; n)$ and $\mathcal{E}_{\text{binary}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}}; n)$, to emphasize that these losses vary with n.

In addition to this and the broad setup as described in Section 3.1 we make a 1-*sparse assumption* on the unknown parameter vector $\boldsymbol{\alpha}^*$, as described below.

Assumption 1 (1-sparse linear model)**.** *Recall that the bi-level ensemble sets $s := n^r$. For some unknown*[7] *$\tau \in \{1, \ldots, s\}$, we assume that $\boldsymbol{\alpha}^* = \frac{1}{\sqrt{\lambda_\tau}} \cdot \mathbf{e}_\tau$, i.e. the parameter vector $\boldsymbol{\alpha}^*$ is 1-sparse.*

Assumption 1 is most useful to for us to derive clean expressions for regression and binary classification error in terms of natural notions of "survival" and "contamination", as detailed subsequently in Section 3.4. While this assumption appears rather strong, it is actually without loss of generality within the bi-level ensemble *for analyzing the performance of minimum-ℓ_2-norm interpolation specifically.* If the true parameter vector $\boldsymbol{\alpha}^*$ has support only within the s favored directions, then we can choose another orthonormal coordinate system in which this $\boldsymbol{\alpha}^*$ is only along the first direction. Because minimum-ℓ_2-norm interpolation does not care about orthonormal coordinate changes and such a change will not change the underlying covariance matrix, we just assume 1-sparsity to capture the representability of the true model by the favored features.

Under Assumption 1, we now show the existence of a regime, corresponding to choice of (p, q, r) above, for which the regression test loss stays prohibitively high, but the binary classification test loss goes to 0 as $n \to \infty$. (We also derive non-asymptotic versions of these results in Section 3.7, but only state the asymptotic results here for brevity.)

Theorem 2. *Assume that the true data generating process is 1-sparse (Assumption 1). For the bi-level covariance matrix model, the limiting binary classification and regression error of the minimum-ℓ_2-norm interpolation (of binary labels and real labels respectively) converge in probability, over the randomness in the training data, as a function of the parameters (p, q, r) in the following way:*

[7] The intuition for this condition, also motivated in prior analyses of minimum-ℓ_2-norm interpolation [105], is that for any reasonable preservation of signal, the true feature needs to be sufficiently preferred, therefore weighted highly.

1. *For $0 \leq q < (1-r)$, we have*

$$\lim_{n\to\infty} \mathcal{E}_{\text{reg}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}; n) = 0,$$
$$\lim_{n\to\infty} \mathcal{E}_{\text{binary}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}}; n) = 0.$$

   In this regime, both regression and binary classification generalize well.

2. *For $(1-r) < q < (1-r) + \frac{(p-1)}{2}$, we have*

$$\lim_{n\to\infty} \mathcal{E}_{\text{reg}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}; n) = 1,$$
$$\lim_{n\to\infty} \mathcal{E}_{\text{binary}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}}; n) = 0.$$

   In this regime, binary classification generalizes well but regression does not.

3. *For $(1-r) + \frac{(p-1)}{2} < q \leq (p-r)$, we have*

$$\lim_{n\to\infty} \mathcal{E}_{\text{reg}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}; n) = 1,$$
$$\lim_{n\to\infty} \mathcal{E}_{\text{binary}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}}; n) = \frac{1}{2}.$$

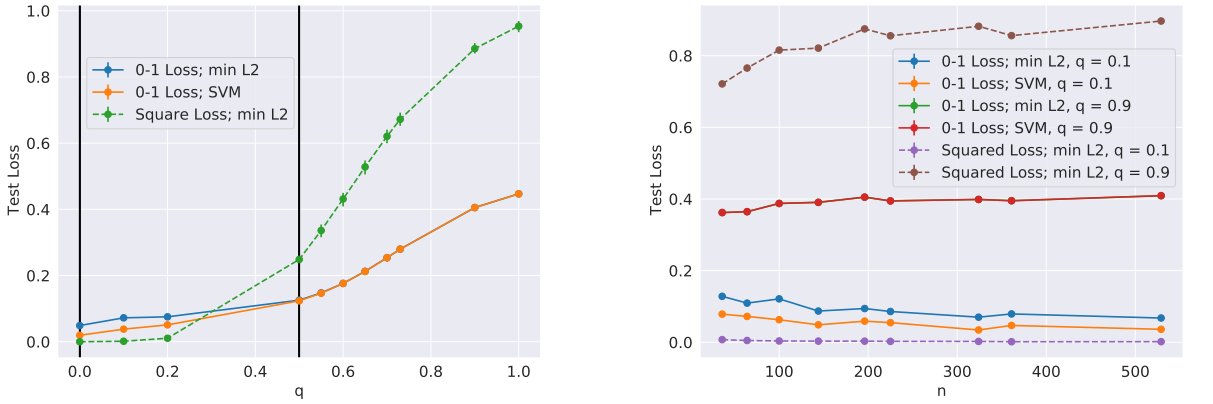
   In this regime, the generalization is poor for both binary classification and regression.

Note that the presence of label noise ν^* does not affect these asymptotic scalings (since $\nu^* < 0.5$).

Figure 3.3(a) shows the evolution of binary classification and regression error as a function of the parameter q, fixing $p = 3/2$ and $r = 1/2$. The binary classification error is plotted for both the SVM and the minimum-ℓ_2-norm interpolation — as we expect from Theorem 1, these are remarkably similar. Figure 3.3(b) shows that the empirical quantities converge to the limiting quantities from Theorem 2. Figure 3.4, visualizes the three asymptotic regimes from Theorem 2 for different fixed values of q. The new regime of principal interest that we have identified is values of $q \in (1-r, 1-r+\frac{p-1}{2})$ for which binary classification generalizes, but regression does not. The entire proof of Theorem 2 is deferred to Sections 3.6 and 3.7, but we briefly illustrate the intuition for this discrepancy between binary classification and regression tasks in Section 3.4. In particular, we will see that good generalization for binary classification requires a far less stringent condition on coefficient recovery than regression.

We now provide some intuition for the scalings described in Theorem 2 for the bi-level ensemble.

Remark 4. *Observe that in this ensemble, regression tasks generalize well iff we have $q < (1-r)$, which is a condition directly related to signal preservation. Recall that for fixed values of (p, r), the parameter q controls the relative ratio of the larger eigenvalues to the smaller eigenvalues (corresponding to unimportant directions). The higher the value of q, the smaller*

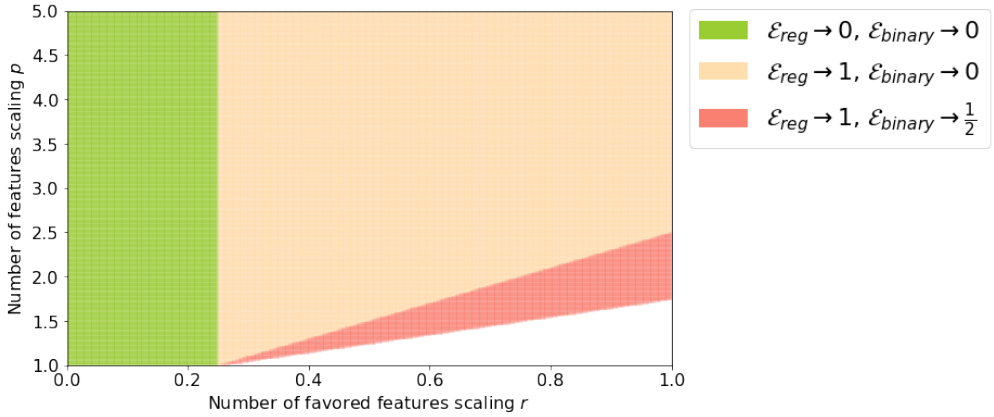


(a) Test loss vs q (b) Test loss vs number of training points

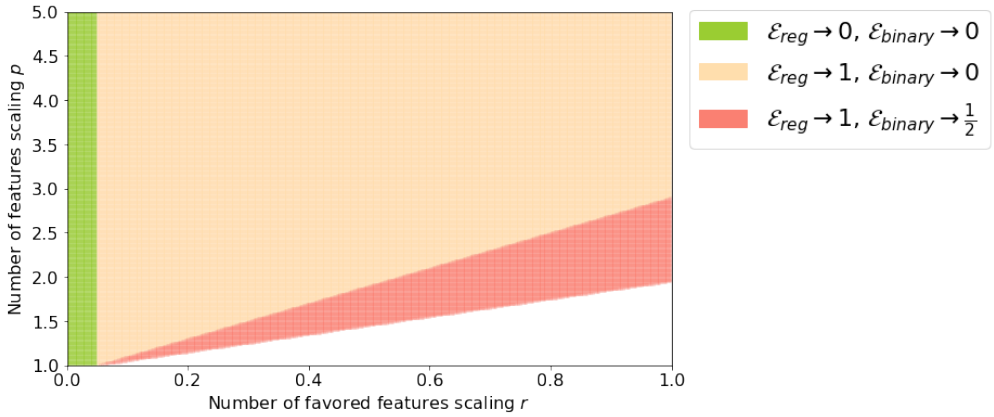
Figure 3.3. Comparison of test binary classification and regression error on solutions obtained by minimizing different choices of training loss on the bi-level ensemble. For both figures, parameters $(p = 3/2, r = 1/2)$ are fixed. On the left, $n = 529, d = 12167$ are fixed. Here, the dashed green curve corresponds to $\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}$ (Equation 3.3b), the orange curve corresponds to $\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}}$ (Equation 3.3a), the solid blue curve corresponds to $\widehat{\boldsymbol{\alpha}}_{\mathsf{SVM}}$ (Equation 3.6), and the black lines demarcate the regimes from Theorem 2. On the right, d varies as $n^{\frac{3}{2}}$.

this ratio, and the harder it is to preserve signal. The results on "benign overfitting" [10] upper bound the contribution of (bounded ℓ_2-norm) pure signal to regression error. This upper bound can also be verified to decay with n iff we have $q \leq (1-r)$. Furthermore, as we already remarked on Definition 7, the bi-level ensemble is designed to always avoid harmful noise overfitting. (We will, however, see in the next remark that the rate of effective noise absorption is important.)

Remark 5. *The regime that we have identified that is of principal interest is intermediate values of q, i.e. $(1-r) < q < (1-r) + \frac{(p-1)}{2}$. This highlights a fascinating role that over-parameterization, in the form of the parameter p, plays in allowing the good generalization of interpolating solutions in binary classification tasks. Recall that the larger the value of p, the larger the total number of features $d = n^p$. Thus, there are several "unimportant directions" in the bi-level ensemble all corresponding to the smaller eigenvalue — which helps in harmless absorption of effective noise. In the proof of Theorem 2, we will identify an explicit mechanism by which having many unimportant directions helps in good generalization for binary classification, even though the signal is not preserved. At a high level, this mechanism constitutes the spreading out of attenuated signal across several features in a relatively "harmless" way, to exhibit minimal influence on classification performance. In fact, this influence is quantified by a notion of "contamination" by falsely discovered features (as we have seen in the previous chapters) that can be directly linked to the contribution of noise overfitting to regression error.*



(a) $q = 0.75$



(b) $q = 0.95$

Figure 3.4. Visualization of the three asymptotic regimes from Theorem 2. In the green region both regression and binary classification generalize well and in the red region neither regression nor binary classification generalize well. There is an intermediate region shown in orange where binary classification generalizes even though regression does not.

Finally, we remark that Theorem 2 provides a connection between binary classification and regression test error when both tasks are solved using the minimum-ℓ_2-norm interpolation, i.e. minimizing the square loss on training data. Since we explicitly linked the minimum-ℓ_2-norm interpolation and the SVM in the preceding Section 3.3, it is natural to ask whether the generalization results in Theorem 2 help us directly compare the SVM for binary classification tasks and the minimum-ℓ_2-norm interpolation for regression tasks. We can indeed do this in a slightly more restricted regime of the bi-level ensemble, described below.

Corollary 1. *Assume that the true data generating process is* 1*-sparse (Assumption 1). Consider the bi-level ensemble with* $p > 2$*. Then, the classification error of the SVM (on binary labels), and the regression error of the minimum-*ℓ_2*-norm interpolation (on real labels),*

converge in probability as follows:

1. *For* $\left(\frac{3}{2} - r\right) < q < (1-r) + \frac{(p-1)}{2}$, *we have*

$$\lim_{n\to\infty} \mathcal{E}_{\text{reg}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}; n) = 1,$$
$$\lim_{n\to\infty} \mathcal{E}_{\text{binary}}(\widehat{\boldsymbol{\alpha}}_{\mathsf{SVM}}; n) = 0.$$

2. *For* $(1-r) + \frac{(p-1)}{2} < q \leq (p-r)$, *we have*

$$\lim_{n\to\infty} \mathcal{E}_{\text{reg}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}; n) = 1,$$
$$\lim_{n\to\infty} \mathcal{E}_{\text{binary}}(\widehat{\boldsymbol{\alpha}}_{\mathsf{SVM}}; n) = \frac{1}{2}.$$

Observe that Corollary 1 directly follows from plugging in the condition required in the bi-level ensemble for all training points usually becoming support vectors (Equation (3.19)), and noting that for $p > 2$, we have

$$(1-r) + \frac{(p-1)}{2} > (1-r) + \frac{1}{2} = \left(\frac{3}{2} - r\right).$$

Importantly, we have identified that even highly overparameterized regimes, in which all training points become support vectors, can yield good generalization for binary classification tasks when the hard-margin SVM is used. Interestingly, we are able to prove good generalization for binary classification even though margin-based generalization bounds are uninformative in sufficiently overparameterized settings (See Section 6 in [104] for more on this).

Path to analysis: Binary classification vs regression test error

The first step to proving Theorem 2 is obtaining clean expressions for both binary classification and regression test error. The 1-sparsity assumption that we have made on the unknown signal enables us to do this as a function of natural quantities corresponding to the preservation of the true feature (*survival*) and the pollution due to false features (*contamination*). If we assume that the real labels are generated by the τ^{th} feature, α^*_τ, then we can define these quantities for any solution $\widehat{\boldsymbol{\alpha}}$. First, as classically observed in statistical signal processing, the estimated coefficient corresponding to the true feature α^*_τ will experience *shrinkage* and be attenuated by a factor that we denote as *survival*. From Assumption 1, we defined $\boldsymbol{\alpha}^* := \frac{1}{\sqrt{\lambda_t}} \cdot \mathbf{e}_\tau$, and so we have

$$\mathsf{SU}(\widehat{\boldsymbol{\alpha}}, \tau) = \frac{\widehat{\boldsymbol{\alpha}}_\tau}{\alpha^*_\tau} = \sqrt{\lambda_\tau}\widehat{\boldsymbol{\alpha}}_\tau \tag{3.20}$$

Second, we have the *false discovery of features*. We measure the effect of this false discovery for prediction on a test point X by a *contamination* term:

$$B = \sum_{j=1, j\neq\tau}^{d} \widehat{\boldsymbol{\alpha}}_j \boldsymbol{\phi}_j(\mathbf{X}). \tag{3.21}$$

Recall that X is random, and the features $\phi(X)$ are zero-mean. Therefore, B is a zero-mean random variable. Accordingly, we can define the standard deviation of the contamination term on a test point as below:

$$\begin{aligned} \mathsf{CN}(\widehat{\boldsymbol{\alpha}}, \tau) &= \sqrt{\mathbb{E}\left[B^2\right]} \\ &= \sqrt{\sum_{j=1, j\neq\tau}^{d} \lambda_j \widehat{\boldsymbol{\alpha}}_j^2}. \end{aligned} \tag{3.22}$$

where the last step follows from the orthogonality of the d features. The ideas of survival and contamination can be related to the classical signal-processing concept of *aliasing*; Figure 1.3 in Section 2.2 provides an illustration.

We state and prove the following proposition, which directly expresses regression and binary classification test loss in terms of these terms.

Proposition 1. *Under the* 1*-sparse noiseless linear model, the regression test loss (excess MSE) is given by:*

$$\mathcal{E}_{\text{reg}}(\widehat{\boldsymbol{\alpha}}) = (1 - \mathsf{SU}(\widehat{\boldsymbol{\alpha}}, \tau))^2 + \mathsf{CN}^2(\widehat{\boldsymbol{\alpha}}, \tau). \tag{3.23}$$

and the binary classification test loss (excess classification error) is given by:

$$\mathcal{E}_{\text{binary}}(\widehat{\boldsymbol{\alpha}}) = \frac{1}{2} - \frac{1}{\pi}\mathsf{tan}^{-1}\left(\frac{\mathsf{SU}(\widehat{\boldsymbol{\alpha}}, \tau)}{\mathsf{CN}(\widehat{\boldsymbol{\alpha}}, \tau)}\right). \tag{3.24}$$

We can think of the quantity $\mathsf{SU}(\widehat{\boldsymbol{\alpha}}, \tau)/\mathsf{CN}(\widehat{\boldsymbol{\alpha}}, \tau)$ *as the effective "signal-to-noise ratio" for binary classification problems.*

Proof. We first prove Equation (3.23). Recall that for any estimator $\widehat{\boldsymbol{\alpha}}$, the excess MSE is given by

$$\begin{aligned} \mathcal{E}_{\text{reg}}(\widehat{\boldsymbol{\alpha}}) :&= \mathbb{E}[(\langle \phi(X),\, \boldsymbol{\alpha}^* - \widehat{\boldsymbol{\alpha}} \rangle)^2] \\ &= \sum_{j=1}^{d} \lambda_j (\alpha_j^* - \widehat{\boldsymbol{\alpha}}_j)^2, \end{aligned}$$

and then substituting in the 1-sparse Assumption 1 gives us Equation (3.23).

Next, we prove Equation (3.24). Since $\phi(X) = \mathbf{\Sigma}^{1/2}\mathbf{W}$ for $\mathbf{W} = (W_1, \ldots, W_d) \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, we can write $\phi(X)^\top \boldsymbol{\alpha}^* = W_\tau$ and $\phi(X)^\top \widehat{\boldsymbol{\alpha}} = \sum_{j=1}^d \sqrt{\lambda_j} W_j \widehat{\boldsymbol{\alpha}}_j$. Thus, the excess binary classification error of $\widehat{\boldsymbol{\alpha}}$ is given by

$$\mathcal{E}_{\text{binary}}(\widehat{\boldsymbol{\alpha}}) = \mathbb{P}\left(\phi(X)^\top \widehat{\boldsymbol{\alpha}} \phi(X)^\top \boldsymbol{\alpha}^* \leq 0\right) = \mathbb{P}\left(\sqrt{\lambda_\tau}\widehat{\boldsymbol{\alpha}}_\tau W_\tau^2 + W_\tau \cdot \sum_{j \neq \tau} \sqrt{\lambda_j}\widehat{\boldsymbol{\alpha}}_j W_j \leq 0\right).$$

Now, the random sum $\sum_{j\neq\tau} \sqrt{\lambda_j}\widehat{\boldsymbol{\alpha}}_j W_j$ has a Gaussian distribution with mean zero and variance $\mathsf{CN}(\widehat{\boldsymbol{\alpha}}, \tau)^2$. Since the $\{W_j\}_{j=1}^d$ are independent, the binary classification test error of $\widehat{\boldsymbol{\alpha}}$ is the probability of the following event:

$$\mathsf{SU}(\widehat{\boldsymbol{\alpha}}, \tau) U^2 + U \cdot \mathsf{CN}(\widehat{\boldsymbol{\alpha}}, \tau) V \leq 0,$$

where U and V are independent standard Gaussian random variables. This event is equivalently written as

$$\frac{V}{U} \leq -\frac{\mathsf{SU}(\widehat{\boldsymbol{\alpha}}, \tau)}{\mathsf{CN}(\widehat{\boldsymbol{\alpha}}, \tau)}.$$

Since V/U follows the standard Cauchy distribution with cumulative distribution function $F(t) = \frac{1}{2} + \frac{1}{\pi}\mathsf{tan}^{-1}(t)$, the claim follows. □

Equations (3.23) and (3.24) give us an initial clue as to why binary classification test error can be easier to minimize than regression test error. For the right hand side of Equation (3.23) to be small, we need $\mathsf{SU} \to 1$ to avoid shrinkage, as well as $\mathsf{CN} \to 0$ to avoid contamination. However, for the right hand side of Equation (3.24) to be small, we only require the ratio of contamination to survival to be small (i.e. $\mathsf{CN}/\mathsf{SU} \to 0$). Clearly, the former condition directly implies the latter, showing that binary classification is "easier" than regression[8]. Theorem 2 is proved fully in Sections 3.6 and 3.7 in the following series of steps:

1. Matching (non-asymptotic) upper and lower bounds are proved on both survival and contamination for interpolation of both real and binary labels. The full statements for these bounds are contained in Theorems 3 and 4 in Section 3.6.

2. These bounds are substituted into the bi-level ensemble to get asymptotic scalings for binary classification and regression test error (Section 3.7).

The bulk of the technical work is involved in proving the matching bounds on survival and contamination, i.e. Theorems 3 and 4. Although these results are inspired by the calculations provided in Section 3.2 for the Fourier case, we build on the techniques provided in [10] for Gaussian features, particularly making use of fundamental concentration bounds that were proved on "leave-one-out" matrices in that work. We build on these techniques to sharply

[8]Our decomposition of binary classification error is reminiscent of the decomposition by [50] into the ratio of terms depending on the variance (like contamination) and bias (like survival) respectively. Because our data is Gaussian, Proposition 1 allows an *exact* decomposition.

bound both the "survival" and "contamination" terms, and thus obtain matching upper and lower bounds for the binary classification test error. Crucially, our analysis needs to circumvent issues that stem from effective misspecification in the linear model that arise from the sign operator. While we do not provide a generic analysis of "misspecification noise," we exploit the special misspecification induced by the sign operator in a number of technical equivalents of the aforementioned random matrix concentration results.

We essentially show that this induced misspecification makes no difference, asymptotically, to classification error arising from interpolation from binary labels, and the behavior is essentially the same as though we had instead interpolated the real output. This is another interesting consequence of requiring only the ratio $\frac{\mathsf{CN}}{\mathsf{SU}} \to 0$, as opposed to the stronger requirements for regression, $\mathsf{CN} \to 0$ and $\mathsf{SU} \to 1$. We will see in Section 3.7 that in the asymptotic limit $n \to \infty$, interpolation of binary *noiseless* labels attenuates the signal by a factor exactly equal to $\sqrt{\frac{2}{\pi}}$. This also corresponds to the attenuation factor of signal that has been traditionally been observed as a result of 1-bit quantization applied before a matched filter[9] [142, 24]. Since this factor is strictly positive, it does not affect the asymptotic binary classification error.

In fact, the non-asymptotic scalings of survival and contamination terms are unaffected even by non-zero label noise on binary classification training data, provided that the label noise still preserves non-trivial information about the signal. The survival is further attenuated by a non-zero factor of $(1-2\nu^*)$, which is strictly positive as long as $\nu^* < 1/2$. Observe that this is equivalent to a hypothetical scenario where the binary labels take on "shrunk" values $\{-(1-2\nu^*), (1-2\nu^*)\}$ instead of the usual $\{-1, 1\}$. As long as $\nu^* < 1/2$, the magnitude of the labels is strictly non-zero and so the labels still provide useful information for binary classification.

Finally, it is natural to ask how fundamental our assumptions of Gaussianity on data and bi-level covariance structure are to our main generalization result (Theorem 2). We chose the bi-level ensemble to illustrate the separation between binary classification and regression in the cleanest possible way. However, Theorems 3 and 4 do provide non-asymptotic expressions for survival and contamination for *arbitrary* covariance matrices. In principle, these expressions can be plugged into Proposition 1 to get upper and lower bounds on binary classification error for arbitrary covariance matrices. Further, the analysis of benign overfitting in linear regression [10, 105] extends to sub-Gaussian features. In the same spirit, we can show that the results — including the existence of the intermediate regime, in which binary classification works but regression does not — extend to a weaker assumption of *independence* and sub-Gaussianity on the underlying features. This extension uses an argument similar to the Fourier-case argument given in Section 3.2 but requires a more direct treatment of the approximation error arising from misspecification. Our results *do not* extend to kernel settings, where there can be complex dependencies among the (infinite-dimensional) features. This is an important direction for future work.

[9] Recall that [105] naturally connected matched filtering to minimum-ℓ_2-norm interpolation.

3.5 Appendix: Additional notation for proofs

We consider zero-mean Gaussian featurization, i.e. $\phi(X_i) = \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. For ease of exposition, we consider $\mathbf{\Sigma}$ to be diagonal[10]. Corresponding to a given index $\tau \in \{1, \ldots, d\}$, we define the "leave-one-out" matrix $\mathbf{\Sigma}_{-\tau}$ whose eigenvalues are given by: $\mu_j(\mathbf{\Sigma}_{-\tau}) = \widetilde{\lambda}_j$ for $j \in \{1, \ldots, d-1\}$. The relation between the spectrum $\{\widetilde{\lambda}_j\}_{j=1}^{d-1}$ and $\{\lambda_j\}_{j=1}^{d}$ is given by

$$\widetilde{\lambda}_j = \begin{cases} \lambda_j, & j < \tau \\ \lambda_{j+1}, & j \geq \tau \end{cases}. \tag{3.25}$$

Consider $\{\mathbf{z}_j\}_{j=1}^{d}$ i.i.d. with $\mathbf{z}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I_n})$. Observe that we can write effective Gram matrices corresponding to the full as well as the "leave-one-out" spectrum of the covariance matrix:

$$\mathbf{A} = \sum_{j=1}^{d} \lambda_j \mathbf{z}_j \mathbf{z}_j^\top = \mathbf{\Phi}_{\mathsf{train}} \mathbf{\Phi}_{\mathsf{train}}^\top, \qquad \mathbf{A}_{-\tau} = \sum_{j=1, j\neq\tau}^{d} \lambda_j \mathbf{z}_j \mathbf{z}_j^\top. \tag{3.26}$$

Using Equation (3.25), we can also express the "leave-one-out" Gram matrix $\mathbf{A}_{-\tau}$ as follows:

$$\mathbf{A}_{-\tau} = \sum_{j=1}^{d-1} \widetilde{\lambda}_j \mathbf{z}_j \mathbf{z}_j^\top. \tag{3.27}$$

We will use both of the above expressions for the leave-one-out matrix $\mathbf{A}_{-\tau}$ in our analysis.

3.6 Appendix: Proof of Theorem 2-Bounds on survival and contamination

In this section, we obtain a general, non-asymptotic characterization of binary classification (and regression) error by bounding survival and contamination terms. As described in Section 3.4, this is then plugged into the expressions in Proposition 1 to prove Theorem 2.

First, we define shorthand notation that is useful for this section, in addition to the notation already defined in Section 3.5. For ease of notation, we denote the survival and contamination factors under the 1-sparse model for the case where we interpolate binary labels as

$$\mathsf{SU}_b(\tau) = \mathsf{SU}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}}, \tau), \quad \mathsf{CN}_b(\tau) = \mathsf{CN}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}}, \tau),$$

and for the case where we interpolate real output as

$$\mathsf{SU}_r(\tau) = \mathsf{SU}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}, \tau), \quad \mathsf{CN}_r(\tau) = \mathsf{CN}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}, \tau).$$

[10]This is without loss of generality: if $\mathbf{\Sigma}$ were not diagonal, we could first do a coordinate transformation to the basis of the eigenvectors of $\mathbf{\Sigma}$.

Finally, for a given index $\tau \in \{1, \ldots, d\}$, we denote as shorthand $\mathbf{z}_\tau := \mathbf{Z}_{\mathsf{train}}$. It is easy to verify that $\mathbf{z}_\tau \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$ under the 1-sparse Assumption 1. We also denote $\mathbf{y}_\tau := \mathbf{Y}_{\mathsf{train}}$. Recall that we consider the possibility of label noise probability equal to ν^*: from the generative model defined in Equation (3.1), we have

$$y_{\tau,i} = \begin{cases} \mathsf{sgn}(z_{\tau,i}) \text{ with probability } (1-\nu^*) \\ -\mathsf{sgn}(z_{\tau,i}) \text{ with probability } \nu^*. \end{cases} \tag{3.28}$$

for every $i \in \{1, \ldots, n\}$. Finally, for a given positive semi-definite matrix $\mathbf{M} \in \mathbb{R}^{d \times d}$ and a given index
$k \in \{0, \ldots, (d-1)\}$, we define the *effective rank*

$$r_k(\mathbf{M}) := \frac{\sum_{\ell > k} \mu_\ell(\mathbf{M})}{\mu_{k+1}(\mathbf{M})}.$$

Recall that this is the precisely the definition of the first effective rank in [10], which dictates the contribution of pure signal to regression test error incurred by the minimum-ℓ_2-norm interpolation.

Bounds on survival and contamination

The notions of survival and contamination were first introduced in [105], and characterized there with equality for Fourier featurization on regularly spaced training data. Here, we characterize these quantities for Gaussian features. We state our upper and lower bounds on survival and contamination respectively for two cases — when the output being interpolated is binary, and when the output being interpolated is real. We start with upper and lower bounds on the survival factor.

Theorem 3 (Upper and lower bounds on survival)**.** *There exist universal positive constants (b, b_2, c, c_3, c_4) (that do not depend on parameters $(n, d, k, \mathbf{\Sigma})$) such that if $r_k(\mathbf{\Sigma}) \geq bn$ and $r_k(\mathbf{\Sigma}_{-\tau}) \geq b_2 n$, we have the following characterizations of the survival factor for any $k \geq \tau$:*

1. ***Interpolation of binary labels:*** *The minimum-ℓ_2-norm interpolation of binary labels, i.e. $\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}}$, satisfies each of*

$$\mathsf{SU}_b(\tau) \geq \sqrt{\frac{2}{\pi}} \cdot (1-2\nu^*) \cdot \frac{\lambda_\tau \left(\frac{(n-k)}{c\widetilde{\lambda}_{k+1} r_k(\mathbf{\Sigma}_{-\tau})} - \frac{c_3 n^{3/4}}{\lambda_{k+1} r_k(\mathbf{\Sigma})} \right)}{1 + \lambda_\tau \left(\frac{cn}{\widetilde{\lambda}_{k+1} r_k(\mathbf{\Sigma}_{-\tau})} + \frac{c_4 n^{3/4}}{\lambda_{k+1} r_k(\mathbf{\Sigma})} \right)}, \text{ and} \tag{3.29a}$$

$$\mathsf{SU}_b(\tau) \leq \sqrt{\frac{2}{\pi}} \cdot (1-2\nu^*) \cdot \frac{\lambda_\tau \left(\frac{cn}{\widetilde{\lambda}_{k+1} r_k(\mathbf{\Sigma}_{-\tau})} + \frac{c_3 n^{3/4}}{\lambda_{k+1} r_k(\mathbf{\Sigma})} \right)}{1 + \lambda_\tau \left(\frac{(n-k)}{c\widetilde{\lambda}_{k+1} r_k(\mathbf{\Sigma}_{-\tau})} - \frac{c_4 n^{3/4}}{\lambda_{k+1} r_k(\mathbf{\Sigma})} \right)} \tag{3.29b}$$

with probability at least $(1 - 3e^{-\sqrt{n}} - 2e^{-\frac{n}{c}})$ over the randomness in the training data $\{X_i, Y_i\}_{i=1}^n$.

2. ***Interpolation of real output:*** *The minimum-ℓ_2-norm interpolation of real output, i.e. $\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}$, satisfies each of*

$$\mathsf{SU}_r(\tau) \geq \frac{1}{1 + \frac{1}{\lambda_\tau\left(\frac{(n-k)}{c\widetilde{\lambda}_{k+1} r_k(\boldsymbol{\Sigma}_{-\tau})} - \frac{c_4 n^{\frac{3}{4}}}{\lambda_{k+1} r_k((\boldsymbol{\Sigma})}\right)}}, \text{ and} \tag{3.30a}$$

$$\mathsf{SU}_r(\tau) \leq \frac{1}{1 + \frac{1}{\lambda_\tau\left(\frac{cn}{\widetilde{\lambda}_{k+1} r_k(\boldsymbol{\Sigma}_{-\tau})} + \frac{c_4 n^{\frac{3}{4}}}{\lambda_{k+1} r_k((\boldsymbol{\Sigma})}\right)}} \tag{3.30b}$$

with probability at least $(1 - 2e^{-\sqrt{n}} - 2e^{-\frac{n}{c}})$ over the randomness in the training data $\{X_i, Y_i\}_{i=1}^n$.

We will see subsequently (in Section 3.7) that the survival bounds, whether binary labels or real output are interpolated, are matching in their dependence on n up to constants. We now state our characterization of the contamination factor.

Theorem 4 (Upper and lower bounds on contamination)**.** *There exist universal positive constants $b_2, c_5, c_6, c_7, c_8, c_9$ (that do not depend on parameters $(n, d, k, \boldsymbol{\Sigma})$) such that if $0 \leq k \leq n/c_5$ and $r_k(\boldsymbol{\Sigma}_{-\tau}) \geq b_2$, the following characterizations of the contamination factor hold for any choice of $\ell \leq k$:*

1. ***Interpolation of binary labels:*** *Provided that $n \geq c_6$, the minimum-ℓ_2-norm interpolation of binary labels, i.e. $\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}}$, satisfies each of*

$$\mathsf{CN}_b(\tau) \leq c_7 \cdot \sqrt{\left(\frac{\ell}{n} + n \cdot \frac{\sum_{j>\ell} \widetilde{\lambda}_j^2}{\left(\sum_{j>k} \widetilde{\lambda}_j\right)^2}\right) \cdot \ln n \cdot (1 + \mathsf{SU}_b(\tau)^2)}, \text{ and} \tag{3.31a}$$

$$\mathsf{CN}_b(\tau) \geq \sqrt{n} \cdot \frac{\sqrt{r_k\left(\boldsymbol{\Sigma}_{-\tau}^2\right) \cdot \widetilde{\lambda}_{k+1}^2}}{c_9\left(\sum_{j=1}^{d} \lambda_j + \lambda_1 n\right)} \tag{3.31b}$$

almost surely for any realization of the random quantity $\mathsf{SU}_b(\tau)$, and with probability at least $\left(1 - \frac{3}{n}\right)$ and $(1 - 2e^{-\frac{n}{c_8}})$ respectively over the randomness in the training data $\{X_i, Y_i\}_{i=1}^n$.

2. ***Interpolation of real output:*** *Provided that $n \geq c_6$, the minimum-ℓ_2-norm interpo-*

lation of real output, i.e. $\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}$*, satisfies each of*

$$\mathsf{CN}_r(\tau) \leq c_7 \left|1 - \mathsf{SU}_r(\tau)\right| \cdot \sqrt{\left(\frac{l}{n} + n \cdot \frac{\sum_{j>l} \widetilde{\lambda}_j^2}{\left(\sum_{j>k} \widetilde{\lambda}_j\right)^2}\right) \cdot \ln n}, \text{ and} \tag{3.32a}$$

$$\mathsf{CN}_r(\tau) \geq \sqrt{n(1-\delta)} \cdot \frac{\sqrt{r_k\left(\boldsymbol{\Sigma}_{-\tau}^2\right) \cdot \widetilde{\lambda}_{k+1}^2}}{c_9 \left(\sum_{j=1}^{d} \lambda_j + \lambda_1 n\right)} \tag{3.32b}$$

almost surely for any realization of the random quantity $\mathsf{SU}_b(\tau)$*, and with probability at least* $\left(1 - \frac{2}{n}\right)$ *and* $(1 - 2e^{-\frac{n}{c_8}} - e^{-n\delta^2})$ *respectively over the randomness in the training data* $\{X_i, Y_i\}_{i=1}^n$*.*

Observe that the high-probability characterizations of contamination in Theorem 4 themselves hold almost surely for every realization of the respective survival factors for binary and real interpolation, which are random variables. In Section 3.7, these expressions will be used together (with a simple union bound) with the matching high-probability characterization of survival factor in Theorem 3. Unlike for the case of survival, the upper and lower bounds for contamination are not necessarily matching — however, as we will see in Section 3.7, they turn out to match for all parameterizations of the bi-level ensemble.

As a final remark, in both theorem statements, the only randomness over which all probabilities are taken is solely in the training data $\{X_i, Y_i\}_{i=1}^n$. Further, all universal positive constants are taken to be independent of the parameters $(n, d, k, \boldsymbol{\Sigma})$, which entirely describe the problem. In the proofs of Theorems 3 and 4, we will follow these conventions unless specified otherwise.

Background lemmas

We begin our proofs of Theorems 3 and 4 by stating lemmas that serve as background for our analysis. The first lemma is from [10].

Lemma 1. ***Concentration of eigenvalues, Lemmas 9 and 10 in [10]*** *There exist universal positive constants* (b, c) *such that:*

1. *For any* $k \geq 0$ *such that* $r_k(\boldsymbol{\Sigma}) \geq bn$*, we have*

$$\frac{1}{c}\lambda_{k+1} r_k(\boldsymbol{\Sigma}) \leq \mu_n\left(\mathbf{A}\right) \leq \mu_1\left(\mathbf{A}\right) \leq c\left(\sum_{j=1}^{d} \lambda_j + \lambda_1 n\right) \text{ and} \tag{3.33}$$

$$\mu_{k+1}(\mathbf{A}) \leq c\lambda_{k+1} r_k(\boldsymbol{\Sigma}) \tag{3.34}$$

with probability at least $(1 - 2e^{-\frac{n}{c}})$ *over the random matrix* $\mathbf{A}$*.*

2. For any $k \geq \tau$ such that $r_k(\mathbf{\Sigma}) \geq bn$, we have

$$\frac{1}{c}\lambda_{k+1} r_k(\mathbf{\Sigma}) \leq \mu_n\left(\mathbf{A}_{-\tau}\right) \leq \mu_1\left(\mathbf{A}_{-\tau}\right) \leq c\left(\sum_{j=1}^{d} \lambda_j + \lambda_1 n\right) \tag{3.35}$$

with probability at least $(1 - 2e^{-\frac{n}{c}})$ over the random matrix $\mathbf{A}_{-\tau}$.

Further, as corollaries to the above, we have the following statements:

1. For any $k \geq 0$ such that $r_k(\mathbf{\Sigma}) \geq bn$, we have

$$\frac{1}{c\left(\sum_{j=1}^{d} \lambda_j + \lambda_1 n\right)} \leq \mu_n\left(\mathbf{A}^{-1}\right) \leq \mu_1\left(\mathbf{A}^{-1}\right) \leq \frac{c}{\lambda_{k+1} r_k(\mathbf{\Sigma})} \tag{3.36}$$

with probability at least $(1 - 2e^{-\frac{n}{c}})$ over the random matrix $\mathbf{A}$.

2. For any $k \geq \tau$ such that $r_k(\mathbf{\Sigma}) \geq bn$, we have

$$\frac{1}{c\left(\sum_{j=1}^{d} \lambda_j + \lambda_1 n\right)} \leq \mu_n\left(\mathbf{A}_{-\tau}^{-1}\right) \leq \mu_1\left(\mathbf{A}_{-\tau}^{-1}\right) \leq \frac{c}{\lambda_{k+1} r_k(\mathbf{\Sigma})} \tag{3.37}$$

with probability at least $(1 - 2e^{-\frac{n}{c}})$ over the random matrix $\mathbf{A}_{-\tau}$.

Note that using Equation (3.27) to express $\mathbf{A}_{-\tau}$, we can rewrite the bounds in the above lemma in terms of the quantities $\mathbf{\Sigma}_{-\tau}$ and $\widetilde{\lambda}_j$. In particular, it follows that each of

$$\frac{1}{c}\widetilde{\lambda}_{k+1} r_k(\mathbf{\Sigma}_{-\tau}) \leq \mu_n\left(\mathbf{A}_{-\tau}\right) \leq \mu_1\left(\mathbf{A}_{-\tau}\right) \leq c\left(\sum_{j=1}^{d-1} \widetilde{\lambda}_j + \widetilde{\lambda}_1 n\right) \text{ and} \tag{3.38a}$$

$$\frac{1}{c\left(\sum_{j=1}^{d-1} \widetilde{\lambda}_j + \widetilde{\lambda}_1 n\right)} \leq \mu_n\left(\mathbf{A}_{-\tau}^{-1}\right) \leq \mu_1\left(\mathbf{A}_{-\tau}^{-1}\right) \leq \frac{c}{\widetilde{\lambda}_{k+1} r_k(\mathbf{\Sigma}_{-\tau})}. \tag{3.38b}$$

holds with probability at least $(1 - 2e^{-\frac{n}{c}})$. We will also apply Equation (3.34) with $\mathbf{A}_{-\tau}$ instead of $\mathbf{A}$, and use the corresponding condition $r_k(\mathbf{\Sigma}) \geq b_2 n$.
The next lemma is the Hanson-Wright inequality, which shows that the quadratic form of a (sub)-Gaussian random vector concentrates around its expectation.

Lemma 2. ***Hanson-Wright inequality [124]*** *Let* $\mathbf{z}$ *be a random vector composed of i.i.d. random variables that are zero mean and sub-Gaussian with parameter at most* 1*. Then, there exists universal constant* $c > 0$ *such that for any positive semi-definite matrix* $\mathbf{M}$ *and for every* $t \geq 0$*, we have*

$$\mathbb{P}\left[|\mathbf{z}^\top \mathbf{M}\mathbf{z} - \mathbb{E}[\mathbf{z}^\top \mathbf{M}\mathbf{z}]| > \tau\right] \leq 2\exp\left\{-c\min\left\{\frac{t^2}{||\mathbf{M}||_{\mathsf{F}}^2}, \frac{t}{||\mathbf{M}||_{\mathsf{op}}}\right\}\right\}.$$

We will apply this inequality in two ways. First, we will note that $||\mathbf{M}||_{\mathsf{F}}^2 \leq n||\mathbf{M}||_{\mathsf{op}}^2$ and substitute $t := c_1||\mathbf{M}||_{\mathsf{op}} \cdot n^{3/4}$ (where $c_1^2 = \frac{1}{c}$) to get

$$|\mathbf{z}^\top \mathbf{M}\mathbf{z} - \mathbb{E}[\mathbf{z}^\top \mathbf{M}\mathbf{z}]| \leq c_1||\mathbf{M}||_{\mathsf{op}} \cdot n^{3/4} \tag{3.39}$$

with probability at least $(1 - 2e^{-\sqrt{n}})$. Second, we will note that $||\mathbf{M}||_{\mathsf{op}} \leq \mathsf{tr}(\mathbf{M})$ and moreover, $||\mathbf{M}||_{\mathsf{F}}^2 = \mathsf{tr}(\mathbf{M}^2) \leq (\mathsf{tr}(\mathbf{M}))^2$. Then, substituting $t := \frac{1}{c} \cdot \mathsf{tr}(\mathbf{M}) \cdot (\ln n)$, we get

$$\mathbf{z}^\top \mathbf{M}\mathbf{z} \leq \mathbb{E}[\mathbf{z}^\top \mathbf{M}\mathbf{z}] + \frac{1}{c} \cdot \mathsf{tr}(\mathbf{M}) \cdot (\ln n) \leq \left(1 + \frac{1}{c}\right) \cdot \mathsf{tr}(\mathbf{M}) \cdot (\ln n) \tag{3.40}$$

with probability at least $(1 - \frac{1}{n})$. Finally, note that all probabilities are only over the random vector $\mathbf{z}$. We will frequently apply Lemma 2 as a high-probability statement conditioned on the realization of a *random*, almost surely positive semi-definite matrix $\mathbf{M}$ which is independent of $\mathbf{z}$.

Finally, the following lemma bounds the squared norm of a Gaussian random vector by a standard tail bound on chi-squared random variables for e.g. see Chapter 2 of [144], stated for completeness.

Lemma 3. *Let* $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$*. Then, for any* $\delta \in (0, 1)$*, we have*

$$n(1 - \delta) \leq \|\mathbf{z}\|_2^2 \leq n(1 + \delta) \tag{3.41}$$

with probability at least $(1 - 2e^{-n\delta^2})$*.*

Proof of Theorem 3

We first prove Theorem 3, i.e. upper and lower bounds on survival when binary labels *or* real output are interpolated. We start with the slightly more difficult case of interpolation of binary labels (Equations (3.29a) and (3.29b)).

Interpolation of binary labels

Recall that, by Assumption 1, we have $\alpha_t^* = \frac{1}{\sqrt{\lambda_\tau}}$. A standard argument based on Moore-Penrose pseudoinverse calculations shows that $\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}} = \boldsymbol{\Phi}_{\mathsf{train}}^\top (\boldsymbol{\Phi}_{\mathsf{train}} \boldsymbol{\Phi}_{\mathsf{train}}^\top)^{-1} \mathbf{Y}_{\mathsf{train}}$. We get

$$\begin{aligned}
\mathsf{SU}_b(\tau) &= \frac{\widehat{\boldsymbol{\alpha}}_{\tau,2,\mathsf{binary}}}{\alpha_\tau^*} \\
&= \sqrt{\lambda_\tau} \widehat{\boldsymbol{\alpha}}_{\tau,2,\mathsf{binary}} \\
&= \sqrt{\lambda_\tau} \mathbf{e}_\tau^\top \boldsymbol{\Phi}_{\mathsf{train}}^\top (\boldsymbol{\Phi}_{\mathsf{train}} \boldsymbol{\Phi}_{\mathsf{train}}^\top)^{-1} \mathbf{Y}_{\mathsf{train}} \\
&= \lambda_\tau \mathbf{z}_\tau^\top \mathbf{A}^{-1} \mathbf{y}_\tau,
\end{aligned}$$

where $\mathbf{z}_\tau, \mathbf{y}_\tau$ are as defined at the beginning of Section 3.6, and $\mathbf{A} = \boldsymbol{\Phi}_{\mathsf{train}} \boldsymbol{\Phi}_{\mathsf{train}}^\top = \sum_{j=1}^{d} \lambda_j \mathbf{z}_j \mathbf{z}_j^\top$ is the Gram matrix defined in Section 3.5. Next, we use the fact that,

$$\mathbf{A}_{-\tau} = \sum_{j=1, j \neq \tau}^{d} \lambda_j \mathbf{z}_j \mathbf{z}_j^\top$$

along with the Sherman-Morrison-Woodbury identity to get

$$\begin{aligned}
\mathbf{A}^{-1} &= (\lambda_\tau \mathbf{z}_\tau \mathbf{z}_\tau^\top + \mathbf{A}_{-\tau})^{-1} \\
&= \mathbf{A}_{-\tau}^{-1} - \frac{\lambda_\tau \mathbf{A}_{-\tau}^{-1} \mathbf{z}_\tau \mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1}}{1 + \lambda_\tau \mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1} \mathbf{z}_\tau},
\end{aligned} \tag{3.42}$$

Substituting this into the expression for $\mathsf{SU}_b(\tau)$, we obtain

$$\begin{aligned}
\mathsf{SU}_b(\tau) &= \lambda_\tau \mathbf{z}_\tau^\top \mathbf{A}^{-1} \mathbf{y}_\tau \\
&= \lambda_\tau \mathbf{z}_\tau^\top \left(\mathbf{A}_{-\tau}^{-1} - \frac{\lambda_\tau \mathbf{A}_{-\tau}^{-1} \mathbf{z}_\tau \mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1}}{1 + \lambda_\tau \mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1} \mathbf{z}_\tau} \right) \mathbf{y}_\tau \\
&= \frac{\lambda_\tau \mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1} \mathbf{y}_\tau (1 + \lambda_\tau \mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1} \mathbf{z}_\tau) - \lambda_\tau \mathbf{z}_\tau \lambda_\tau \mathbf{A}_{-\tau}^{-1} \mathbf{z}_\tau \mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1} \mathbf{y}_\tau}{1 + \lambda_\tau \mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1} \mathbf{z}_\tau} \\
&= \frac{\lambda_\tau \mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1} \mathbf{y}_\tau}{1 + \lambda_\tau \mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1} \mathbf{z}_\tau}.
\end{aligned} \tag{3.43}$$

Adding and subtracting terms to the numerator, we get

$$\mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1} \mathbf{y}_\tau = \frac{1}{4} \left((\mathbf{z}_\tau + \mathbf{y}_\tau)^\top \mathbf{A}_{-\tau}^{-1} (\mathbf{z}_\tau + \mathbf{y}_\tau) - (\mathbf{z}_\tau - \mathbf{y}_\tau)^\top \mathbf{A}_{-\tau}^{-1} (\mathbf{z}_\tau - \mathbf{y}_\tau) \right).$$

Because of the "leave-one-out" property, note that $\mathbf{A}_{-\tau}^{-1}$ is independent of $\{\mathbf{z}_\tau, \mathbf{y}_\tau\}$. Also note that $\mathbf{A}_{-\tau}^{-1}$ is almost surely positive semidefinite. Thus, we can upper *and* lower bound the

numerator of Equation (3.43) around its expectation using the Hanson-Wright inequality. First, we calculate the conditional expectation:

$$\begin{aligned}\mathbb{E}\left[\mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1}\mathbf{y}_\tau \middle| \mathbf{A}_{-\tau}^{-1}\right] &= \mathbb{E}\left[\mathsf{tr}(\mathbf{A}_{-\tau}^{-1}\mathbf{y}_\tau\mathbf{z}_\tau^\top) \middle| \mathbf{A}_{-\tau}^{-1}\right] \\ &= \mathsf{tr}\left(\mathbf{A}_{-\tau}^{-1}\cdot\mathbb{E}\left[\mathbf{y}_\tau\mathbf{z}_\tau^\top\right]\right).\end{aligned}$$

Recalling the expression for $\mathbf{y}_\tau$ from Equation (3.28), a simple calculation yields that

$$\begin{aligned}\mathbb{E}\left[\mathbf{y}_\tau\mathbf{z}_\tau^\top\right] &= \mathbb{E}\left[y_{\tau,1}z_{\tau,1}^\top\right]\cdot\mathbf{I}_n \\ &= \left((1-\nu^*)\mathbb{E}\left[\mathsf{sgn}(z_{\tau,1})z_{\tau,1}^\top\right] + \nu^*\mathbb{E}\left[-\mathsf{sgn}(z_{\tau,1})z_{\tau,1}^\top\right]\right)\cdot\mathbf{I}_n \\ &= (1-2\nu^*)\mathbb{E}\left[\mathsf{sgn}(z_{\tau,1})z_{\tau,1}^\top\right]\cdot\mathbf{I}_n \\ &= (1-2\nu^*)\cdot\sqrt{\frac{2}{\pi}}\cdot\mathbf{I}_n,\end{aligned}$$

where the last step follows because $z_{\tau,1}\sim\mathcal{N}(0,1)$.

Now, we apply Equation (3.39) (the Hanson-Wright inequality) *almost surely* for every realization of the random matrix $\mathbf{A}_{-\tau}^{-1}$, and simultaneously to the quadratic forms $(\mathbf{z}_\tau+\mathbf{y}_\tau)^\top\mathbf{A}_{-\tau}^{-1}(\mathbf{z}_\tau+\mathbf{y}_\tau)$ and $(\mathbf{z}_\tau-\mathbf{y}_\tau)^\top\mathbf{A}_{-\tau}^{-1}(\mathbf{z}_\tau-\mathbf{y}_\tau)$. Thus, we have each of

$$\begin{aligned}\mathbf{z}_\tau^\top\mathbf{A}_{-\tau}^{-1}\mathbf{y}_\tau &\geq \left((1-2\nu^*)\sqrt{\frac{2}{\pi}}\mathsf{tr}(\mathbf{A}_{-\tau}^{-1}) - 2c_1||\mathbf{A}_{-\tau}^{-1}||_{\mathsf{op}}\cdot n^{3/4}\right) \text{ and} \\ \mathbf{z}_\tau^\top\mathbf{A}_{-\tau}^{-1}\mathbf{y}_\tau &\leq \left((1-2\nu^*)\sqrt{\frac{2}{\pi}}\mathsf{tr}(\mathbf{A}_{-\tau}^{-1}) + 2c_1||\mathbf{A}_{-\tau}^{-1}||_{\mathsf{op}}\cdot n^{3/4}\right)\end{aligned}$$

with probability at least $(1-2e^{-\sqrt{n}})$ over the randomness in $\{\mathbf{z}_\tau,\mathbf{y}_\tau\}$. Similarly, to bound the the denominator, we have each of

$$\mathbf{z}_\tau^\top\mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau \geq \mathsf{tr}(\mathbf{A}_{-\tau}^{-1}) - c_1||\mathbf{A}_{-\tau}^{-1}||_{\mathsf{op}}\cdot n^{3/4} \text{ and} \tag{3.44a}$$

$$\mathbf{z}_\tau^\top\mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau \leq \mathsf{tr}(\mathbf{A}_{-\tau}^{-1}) + c_1||\mathbf{A}_{-\tau}^{-1}||_{\mathsf{op}}\cdot n^{3/4} \tag{3.44b}$$

with probability at least $(1-e^{-\sqrt{n}})$ over the randomness in $\{\mathbf{z}_\tau,\mathbf{y}_\tau\}$. Substituting these bounds into Equation (3.43), we get each of

$$\begin{aligned}\mathsf{SU}_b(\tau) &\geq \frac{\lambda_\tau\cdot\left(\sqrt{\frac{2}{\pi}}(1-2\nu^*)\mathsf{tr}(\mathbf{A}_{-\tau}^{-1}) - 2c_1||\mathbf{A}_{-\tau}^{-1}||_{\mathsf{op}}\cdot n^{3/4}\right)}{1+\lambda_\tau\left(\mathsf{tr}(\mathbf{A}_{-\tau}^{-1}) + c_1||\mathbf{A}_{-\tau}^{-1}||_{\mathsf{op}}\cdot n^{3/4}\right)} \text{ and} \\ \mathsf{SU}_b(\tau) &\leq \frac{\lambda_\tau\cdot\left(\sqrt{\frac{2}{\pi}}(1-2\nu^*)\mathsf{tr}(\mathbf{A}_{-\tau}^{-1}) + 2c_1||\mathbf{A}_{-\tau}^{-1}||_{\mathsf{op}}\cdot n^{3/4}\right)}{1+\lambda_\tau\left(\mathsf{tr}(\mathbf{A}_{-\tau}^{-1}) - c_1||\mathbf{A}_{-\tau}^{-1}||_{\mathsf{op}}\cdot n^{3/4}\right)},\end{aligned}$$

with probability at least $(1-3e^{-\sqrt{n}})$ over the randomness in $\{\mathbf{z}_\tau,\mathbf{y}_\tau\}$. It remains to obtain high-probability bounds on the random quantities $\mathsf{tr}(\mathbf{A}_{-\tau}^{-1})$ and $||\mathbf{A}_{-\tau}^{-1}||_{\mathsf{op}}$. Note that we need

both lower bounds and upper bounds on the quantity $\mathsf{tr}(\mathbf{A}_{-\tau}^{-1})$, but we only need an upper bound on the quantity $||\mathbf{A}_{-\tau}^{-1}||_{\mathsf{op}}$.
We assume that we can choose $k \geq \tau$ such that $r_k(\mathbf{\Sigma}) \geq bn$ and $r_k(\mathbf{\Sigma}_{-\tau}) \geq b_2 n$ for universal positive constants (b, b_2). Consider any such choice of k (which in general could depend on (n, d)). First, we use Equation (3.37) from Lemma 1 to upper bound the quantity $||\mathbf{A}_{-\tau}^{-1}||_{\mathsf{op}}$ as

$$||\mathbf{A}_{-\tau}^{-1}||_{\mathsf{op}} = \mu_1(\mathbf{A}_{-\tau}^{-1}) \leq \frac{c}{\lambda_{k+1} r_k(\mathbf{\Sigma})} \tag{3.45}$$

with probability at least $(1 - e^{-\frac{n}{c}})$ over the random matrix $\mathbf{A}$. Next, we turn to the quantity $\mathsf{tr}(\mathbf{A}_{-\tau}^{-1})$. To lower bound this quantity, we notice that

$$\begin{aligned}
\mathsf{tr}(\mathbf{A}_{-\tau}^{-1}) &= \sum_{j=1}^{n} \frac{1}{\mu_j(\mathbf{A}_{-\tau})} \\
&\geq \sum_{j=k}^{n} \frac{1}{\mu_j(\mathbf{A}_{-\tau})} \\
&\geq \frac{(n-k)}{\mu_{k+1}(\mathbf{A}_{-\tau})}.
\end{aligned}$$

Now, from Equation (3.34) in Lemma 1 applied with $\mathbf{A}_{-\tau}$, we have

$$\mu_{k+1}(\mathbf{A}_{-\tau}) \leq c\widetilde{\lambda}_{k+1} r_k(\mathbf{\Sigma}_{-\tau})$$

with probability at least $(1 - e^{-\frac{n}{c}})$ provided that $r_k(\mathbf{\Sigma}_{-\tau}) \geq b_2 n$. This gives us:

$$\mathsf{tr}(\mathbf{A}_{-\tau}^{-1}) \geq \frac{(n-k)}{c\widetilde{\lambda}_{k+1} r_k(\mathbf{\Sigma}_{-\tau})}. \tag{3.46}$$

with probability at least $(1 - e^{-\frac{n}{c}})$. On the other hand, the upper bound on the trace follows simply by

$$\begin{aligned}
\mathsf{tr}(\mathbf{A}_{-\tau}^{-1}) &\leq \frac{n}{\mu_n(\mathbf{A}_{-\tau})} \\
&\leq \frac{cn}{\widetilde{\lambda}_{k+1} r_k(\mathbf{\Sigma}_{-\tau})},
\end{aligned} \tag{3.47}$$

where the last inequality substitutes Equation (3.38a), which again holds with probability at least $(1 - e^{-\frac{n}{c}})$. Noting that the upper bound on $\mathsf{SU}_b(\tau)$ is monotonically increasing in both $\mathsf{tr}(\mathbf{A}_{-\tau}^{-1})$ and $||\mathbf{A}_{-\tau}^{-1}||_{\mathsf{op}}$, and the lower bound on $\mathsf{SU}_b(\tau)$ is monotonically increasing in $\mathsf{tr}(\mathbf{A}_{-\tau}^{-1})$ but decreasing in $||\mathbf{A}_{-\tau}^{-1}||_{\mathsf{op}}$, we can substitute the above bounds on these quantities. This completes our characterization of survival when binary labels are interpolated, with the probability of this characterization lower bounded by taking a union bound over the complement of all the above events. After taking this union bound, the probability of each of the lower bound (Equation (3.29a)) and upper bound (Equation (3.29b)) holding is *at least* $(1 - 3e^{-\sqrt{n}} - 2e^{-\frac{n}{c}})$.

Interpolation of real output

For completeness, we also include the proof of Theorem 3 for the simpler case of interpolation of real-valued output (Equations (3.30a) and (3.30b)). By the same standard argument, we can characterize the minimum-ℓ_2-norm interpolator of real output as $\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}} = \boldsymbol{\Phi}_{\mathsf{train}}^\top(\boldsymbol{\Phi}_{\mathsf{train}}\boldsymbol{\Phi}_{\mathsf{train}}^\top)^{-1}\mathbf{Z}_{\mathsf{train}}$. By a similar argument to the case of binary labels, we have

$$\begin{aligned}\mathsf{SU}_r(\tau) &= \sqrt{\lambda_\tau}\widehat{\boldsymbol{\alpha}}_\tau \\ &= \sqrt{\lambda_\tau}\mathbf{e}_\tau^\top\boldsymbol{\Phi}_{\mathsf{train}}^\top(\boldsymbol{\Phi}_{\mathsf{train}}\boldsymbol{\Phi}_{\mathsf{train}}^\top)^{-1}\mathbf{Z}_{\mathsf{train}} \\ &= \lambda_\tau\mathbf{z}_\tau^\top\mathbf{A}^{-1}\mathbf{z}_\tau.\end{aligned}$$

Again, using the Sherman-Morrison-Woodbury identity, we have

$$\mathbf{A}^{-1} = \mathbf{A}_{-\tau}^{-1} - \frac{\lambda_\tau\mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau\mathbf{z}_\tau^\top\mathbf{A}_{-\tau}^{-1}}{1+\lambda_\tau\mathbf{z}_\tau^\top\mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau},$$

which gives us

$$\begin{aligned}\mathsf{SU}_r(\tau) &= \frac{\lambda_\tau\mathbf{z}_\tau^\top\mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau}{1+\lambda_\tau\mathbf{z}_\tau^\top\mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau} \\ &= \frac{1}{1+\frac{1}{\lambda_\tau\mathbf{z}_\tau^\top\mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau}}.\end{aligned} \tag{3.48}$$

From Equations (3.44a) and (3.44b) above, the following statements each hold with probability at least $(1-e^{-\sqrt{n}})$ over the randomness in $\mathbf{z}_\tau$ and for every realization of the random matrix $\mathbf{A}_{-\tau}^{-1}$:

$$\begin{aligned}&\mathbf{z}_\tau^\top\mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau \geq \mathsf{tr}(\mathbf{A}_{-\tau}^{-1}) - c_2||\mathbf{A}_{-\tau}^{-1}||_{\mathsf{op}}\cdot n^{3/4} \text{ and} \\ &\mathbf{z}_\tau^\top\mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau \leq \mathsf{tr}(\mathbf{A}_{-\tau}^{-1}) + c_2||\mathbf{A}_{-\tau}^{-1}||_{\mathsf{op}}\cdot n^{3/4}.\end{aligned}$$

Here, c_2 is a universal positive constant.

Observe that the right hand side of Equation (3.48) is increasing in the quantity $\mathbf{z}_\tau^\top\mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau$. Thus, substituting the lower bound for $\mathsf{tr}(\mathbf{A}_{-\tau}^{-1})$ from Equation (3.46) and the upper bound for $||\mathbf{A}_{-\tau}^{-1}||_{\mathsf{op}}$ from Equation (3.45) lower bounds the quantity $\mathbf{z}_\tau^\top\mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau$, yielding the lower bound for $\mathsf{SU}_r(\tau)$. Similarly, substituting the upper bound for $\mathsf{tr}(\mathbf{A}_{-\tau}^{-1})$ from Equation (3.47) and the upper bound for $||\mathbf{A}_{-\tau}^{-1}||_{\mathsf{op}}$ from Equation (3.45) upper bounds the quantity $\mathbf{z}_\tau^\top\mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau$, yielding the upper bound for $\mathsf{SU}_r(\tau)$. This completes the proof of Theorem 3. Again, a simple application of the union bound shows that each of the lower bound (Equation (3.30a)) and the upper bound (Equation (3.30b)) hold with probability at least $(1-2e^{-\sqrt{n}}-2e^{-\frac{n}{c}})$.

Proof of Theorem 4

We next prove Theorem 4, i.e. upper and lower bounds on contamination, for the cases of interpolating binary labels and real output. Since the contamination factor is intricately related to the contribution of additive noise to regression test error, the proof primarily consists of refinements of the arguments in [10].

Interpolation of binary labels

We start with a useful set of expressions for the contamination factor in the following lemma. The proof of this lemma is contained in Section 3.8.

Lemma 4. *The contamination of the minimum-ℓ_2-norm interpolation of binary labels, denoted by $\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}}$, can be written in the following two forms:*

$$\mathsf{CN}_b(\tau) = \sqrt{\mathbf{y}_\tau^\top \mathbf{C} \mathbf{y}_\tau}, \tag{3.49a}$$

$$= \sqrt{\widetilde{\mathbf{y}_\tau}^\top \widetilde{\mathbf{C}} \widetilde{\mathbf{y}_\tau}}, \tag{3.49b}$$

where we denote

$$\widetilde{\mathbf{y}_\tau} := \mathbf{y}_\tau - \mathsf{SU}_b(\tau)\mathbf{z}_\tau \ ,$$

$$\mathbf{C} := \mathbf{A}^{-1} \left(\sum_{j=1, j\neq\tau}^{d} \lambda_j^2 \mathbf{z}_j \mathbf{z}_j^\top \right) \mathbf{A}^{-1} \ , \ \textit{and}$$

$$\widetilde{\mathbf{C}} := \mathbf{A}_{-\tau}^{-1} \left(\sum_{j=1, j\neq\tau}^{d} \lambda_j^2 \mathbf{z}_j \mathbf{z}_j^\top \right) \mathbf{A}_{-\tau}^{-1}.$$

We will use the expression in Equation (3.49b) to prove an upper bound on contamination, and the expression in Equation (3.49a) for the lower bound.

Upper bound on $\mathsf{CN}_b(\tau)$

We start with the proof for the upper bound on contamination for interpolation of binary labels (Equation (3.31a)). From Equation (3.49b) in Lemma 4, we have $\mathsf{CN}_b^2(\tau) = \widetilde{\mathbf{y}_\tau}^\top \widetilde{\mathbf{C}} \widetilde{\mathbf{y}_\tau}$. Note that by construction, $\widetilde{\mathbf{C}}$ has no dependence on $\{\mathbf{z}_\tau, \mathbf{y}_\tau\}$ and thus $\widetilde{\mathbf{C}} \perp \widetilde{\mathbf{y}_\tau}$. The next lemma upper bounds the term $\widetilde{\mathbf{y}_\tau}^\top \widetilde{\mathbf{C}} \widetilde{\mathbf{y}_\tau}$ in terms of $\mathsf{tr}(\widetilde{\mathbf{C}})$ and is proved in Section 3.8.

Lemma 5. *There exists universal positive constant c_6 such that when $n \geq c_6$, we have*

$$\widetilde{\mathbf{y}_\tau}^\top \widetilde{\mathbf{C}} \widetilde{\mathbf{y}_\tau} \leq 2 \left(1 + \frac{1}{c}\right) \cdot (1 + \mathsf{SU}_b(\tau)^2) \cdot \mathsf{tr}(\widetilde{\mathbf{C}}) \cdot \ln n$$

almost surely for every realization of the random matrix $\widetilde{\mathbf{C}}$, and with probability at least $\left(1 - \frac{2}{n}\right)$ over the randomness in $\widetilde{\mathbf{y}_\tau}$.

Applying Lemma 5, we get

$$\mathsf{CN}_b^2(\tau) \leq 2 \left(1 + \frac{1}{c}\right) \cdot \mathsf{tr}(\widetilde{\mathbf{C}}) \cdot \ln n \tag{3.50}$$

almost surely for every realization of the random matrix $\widetilde{\mathbf{C}}$, and with probability at least $\left(1 - \frac{2}{n}\right)$ over the randomness in $\widetilde{\mathbf{y}_\tau}$. The next lemma, which is taken from [10], provides a high-probability upper bound on the quantity $\mathsf{tr}(\widetilde{\mathbf{C}})$.

Lemma 6. *(From Lemma 11 in [10]) There exist universal constants* $(b_2, c_5, c_{10} \geq 1)$ *such that whenever* $0 \leq k \leq n/c_5$ *and* $r_k(\mathbf{\Sigma}_{-\tau}) \geq b_2 n$*, we have*

$$\mathsf{tr}(\widetilde{\mathbf{C}}) \leq c_{10} \cdot \left(\frac{l}{n} + n \cdot \frac{\sum_{j>l} \widetilde{\lambda}_j^2}{\left(\sum_{j>k} \widetilde{\lambda}_j\right)^2} \right)$$

for any choice of $l \leq k$*, with probability at least* $(1 - 6e^{-\frac{n}{c_5}})$ *over the randomness in* $\widetilde{\mathbf{C}}$*.*

Substituting the upper bound from Lemmas 6 and into Equation (3.50), and taking the square root on both sides, we have

$$\mathsf{CN}_b(\tau) \leq \sqrt{2\left(1 + \frac{1}{c}\right) \cdot c_{10} \cdot \left(\frac{l}{n} + n \cdot \frac{\sum_{j>l} \widetilde{\lambda}_j^2}{\left(\sum_{j>k} \widetilde{\lambda}_j\right)^2} \right) \cdot (1 + \mathsf{SU}_b(\tau)^2) \cdot \ln n}.$$

with probability at least $\left(1 - \frac{2}{n} - 6e^{-\frac{n}{c_2}}\right)$ over the training data. Taking $c_7 = \sqrt{2\left(1 + \frac{1}{c}\right) c_{10}}$, the upper bound on $\mathsf{CN}_b(\tau)$ in Equation (3.31a) follows. Noting that $\left(1 - \frac{2}{n} - 6e^{-\frac{n}{c_2}}\right) \geq \left(1 - \frac{3}{n}\right)$ for large enough n, this completes the proof of the upper bound.

Lower bound on $\mathsf{CN}_b(\tau)$

Now we move on to the proof for the lower bound on contamination for interpolation of binary labels (Equation (3.31b)). Using Equation (3.49a) from Lemma 4, we get

$$\begin{aligned} \mathsf{CN}_b^2(\tau) &= \mathbf{y}_\tau^\top \mathbf{C} \mathbf{y}_\tau \\ &\geq \mu_n(\mathbf{C}) \|\mathbf{y}_\tau\|_2^2 = n\mu_n(\mathbf{C}). \end{aligned}$$

The next lemma lower bounds the minimum eigenvalue of $\mathbf{C}$ and is proved in Section 3.8.

Lemma 7. *Let* $k \geq 0$ *and* $r_k\left(\mathbf{\Sigma}_{-\tau}^2\right) \geq b_4 n$*. Then, we have*

$$\mu_n(\mathbf{C}) \geq \frac{r_k\left(\mathbf{\Sigma}_{-\tau}^2\right) \cdot \widetilde{\lambda}_{k+1}^2}{c_{11} \cdot c^2 \cdot \left(\sum_{j=1}^{d} \lambda_j + \lambda_1 n \right)^2}$$

with probability at least $(1 - e^{-\frac{n}{c}} - e^{-\frac{n}{c_{11}}})$*.= over the randomness in* $\mathbf{C}$*. Here,* (b_4, c, c_{11}) *are universal positive constants.*

A direct substitution of the above gives us

$$\mathsf{CN}_b(\tau) \geq \sqrt{n} \cdot \frac{\sqrt{r_k\left(\boldsymbol{\Sigma}_{-\tau}^2\right) \cdot \widetilde{\lambda}_{k+1}^2}}{c \cdot \sqrt{c_{11}} \cdot \left(\sum_{j=1}^{d} \lambda_j + \lambda_1 n\right)}$$

with probability at least $(1 - e^{-\frac{n}{c}} - e^{-\frac{n}{c_{11}}})$ over the training data. Taking $c_9 = c\sqrt{c_{11}}$ and c_8 such that $\frac{1}{c_8} = \min(\frac{1}{c}, \frac{1}{c_{11}})$ holds, the lower bound in Equation (3.31b) follows. This completes the characterization of the contamination factor when we interpolate binary labels.

Interpolation of real output

For completeness, we also provide the proof of Theorem 4 for the simpler case of interpolation of real output. We start with a useful set of expressions for the contamination factor in the following lemma. The proof of this lemma is contained in Section 3.8.

Lemma 8. *The contamination of the minimum-ℓ_2-norm interpolator of binary labels, denoted by $\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}$, can be written in the following two forms:*

$$\mathsf{CN}_r(\tau) = \sqrt{\mathbf{z}_\tau^\top \mathbf{C} \mathbf{z}_\tau}, \tag{3.51a}$$

$$= |1 - \mathsf{SU}_r(\tau)| \sqrt{\mathbf{z}_\tau^\top \widetilde{\mathbf{C}} \mathbf{z}_\tau}, \tag{3.51b}$$

where we denote

$$\mathbf{C} = \mathbf{A}^{-1} \left(\sum_{j=1, j\neq\tau}^{d} \lambda_j^2 \mathbf{z}_j \mathbf{z}_j^\top \right) \mathbf{A}^{-1} \text{ , and}$$

$$\widetilde{\mathbf{C}} = \mathbf{A}_{-\tau}^{-1} \left(\sum_{j=1, j\neq\tau}^{d} \lambda_j^2 \mathbf{z}_j \mathbf{z}_j^\top \right) \mathbf{A}_{-\tau}^{-1}.$$

We will use the form in Equation (3.51b) to prove an upper bound on contamination and the form in Equation (3.51a) for the lower bound.

Upper bound on $\mathsf{CN}_r(\tau)$

We start with the proof for the upper bound on contamination for interpolation of real output (Equation (3.32a). From Equation (3.51a) in Lemma 8, we get

$$\mathsf{CN}_r^2(\tau) = (1 - \mathsf{SU}_r(\tau))^2 \mathbf{z}_\tau^\top \widetilde{\mathbf{C}} \mathbf{z}_\tau. \tag{3.52}$$

From Equation (3.68) in Section 3.8 (proof of Lemma 5), we can upper bound the quadratic form $\mathbf{z}_\tau^\top \widetilde{\mathbf{C}} \mathbf{z}_\tau$ as

$$\mathbf{z}_\tau^\top \widetilde{\mathbf{C}} \mathbf{z}_\tau \leq 7\mathsf{tr}(\widetilde{\mathbf{C}}) \ln n$$

with probability at least $\left(1 - \frac{1}{n}\right)$ over the randomness in $\mathbf{z}_\tau$. Then, substituting the upper bound on $\mathsf{tr}(\widetilde{\mathbf{C}})$ from Lemma 6 directly gives us the expression for the upper bound on $\mathsf{CN}_r(\tau)$. Noting again that $\left(1 - \frac{1}{n} - 6e^{-\frac{n}{c_2}}\right) \geq \left(1 - \frac{2}{n}\right)$ for large enough n, this completes the proof for the upper bound.

Lower bound

We conclude this section by proving the lower bound on contamination for interpolation of real output (Equation (3.32b)). We directly apply Equation (3.51a) (from Lemma 8) to get

$$\begin{aligned}
\mathsf{CN}_r^2(\tau) &= \mathbf{z}_\tau^\top \mathbf{C} \mathbf{z}_\tau \\
&\geq \mu_n(\mathbf{C}) \|\mathbf{z}_\tau\|_2^2 \\
&\overset{\text{(i)}}{\geq} n(1-\delta)\mu_n(\mathbf{C})
\end{aligned}$$

with probability at least $(1 - e^{-n\delta^2})$ over the randomness in $\mathbf{z}_\tau$ for any $\delta \in (0, 1)$. Here, inequality (i) follows from the lower bound in Lemma 3. Finally, substituting the lower bound for $\mu_n(\mathbf{C})$ from Lemma 7 gives us the desired expression for the lower bound on $\mathsf{CN}_r(\tau)$. Note that by the union bound, this expression will hold with probability at least $(1 - e^{-n\delta^2} - e^{-\frac{n}{c}} - e^{-\frac{n}{c_{11}}}) = (1 - 2e^{-\frac{n}{c_8}} - e^{-n\delta^2})$ over the randomness in the training data. This completes the proof of Theorem 4. □

3.7 Appendix: Proof of Theorem 2-Implications for bi-level covariance

In this section, we follow the *path to analysis* described in Section 3.4 and prove Theorem 2 for the bi-level ensemble (Definition 7) in the following series of steps:

1. We substitute the spectrum of the bi-level ensemble into Theorems 3 and 4 to get asymptotic expressions for survival and contamination.

2. We substitute these expressions into the expressions for regression and binary classification test loss (Proposition 1) to characterize the regimes for good generalization of binary classification and regression.

For convenience of notation, we consider $\tau = 1$. (Note, however, that the analysis holds for any $1 \leq \tau \leq s$ since the first s eigenvalues of $\mathbf{\Sigma}$ are equal.) Further, to emphasize that

the survival and contamination quantities depend on n, in this section we refer to them as $\mathsf{SU}_b(1;n), \mathsf{CN}_b(1;n), \mathsf{SU}_r(1;n)$, and $\mathsf{CN}_r(1;n)$ for interpolators of binary and real output respectively.

First, we characterize some useful quantities for the bi-level ensemble. Recall that the bi-level ensemble is parameterized by $p > 1$, $0 < q \leq (p-r)$ and $0 < r \leq 1$. We first compute the effective ranks $r_k(\mathbf{\Sigma})$ and $r_k(\mathbf{\Sigma}_{-\tau})$ for two choices of k. First, we have

$$r_s(\mathbf{\Sigma}) = \frac{1}{\frac{(1-a)d}{d-s}} . \frac{(1-a)d}{d-s} . (d-s) = d - s.$$

Substituting $d = n^p$ and $s = n^r$, we have, for sufficiently large n,

$$r_s(\mathbf{\Sigma}) \asymp n^p \gg n. \tag{3.53}$$

Similarly because $1 \leq \tau \leq s$, we have, for sufficiently large n,

$$r_s(\mathbf{\Sigma}_{-\tau}) = d - s - 1 \asymp n^p \gg n. \tag{3.54}$$

Moreover, we get

$$r_0(\mathbf{\Sigma}) = \frac{1}{\frac{ad}{s}} \cdot d = \frac{s}{a} = n^{q+r} \gg n \text{ iff } (q+r) > 1. \tag{3.55}$$

and by a similar argument, provided that $r > 0$, we can show that (for large enough n),

$$r_0(\mathbf{\Sigma}_{-\tau}) = \frac{1}{\frac{ad}{s}} \cdot \left(d - \frac{ad}{s} \right) = \frac{s}{a} - 1 = n^{q+r} - 1 \gg n \text{ iff } (q+r) > 1. \tag{3.56}$$

We will apply Equations (3.53) and (3.54) for bounding survival in general, as well as contamination when we have $q \leq (1-r)$, and Equations (3.55) and (3.56) for bounding contamination when we have $q > (1-r)$. Now, we state and prove our matching upper and lower bounds for survival for the bi-level ensemble.

Lemma 9 (Survival for interpolation of binary labels)**.** *There exist universal positive constants (L_1, U_1, L_2, U_2) such that for sufficiently large n, we have*

$$\mathsf{SU}_b^L(n) \leq \mathsf{SU}_b(1;n) \leq \mathsf{SU}_b^U(n),$$

with probability at least $(1 - 10e^{-\sqrt{n}})$ over the training data $\{X_i, Y_i\}_{i=1}^n$, where we denote

$$\mathsf{SU}_b^L(n) := \begin{cases} \sqrt{\frac{2}{\pi}}(1-2\nu^*) \left(1 + L_1 n^{q-(1-r)}\right)^{-1}, & q < (1-r) \\ \sqrt{\frac{2}{\pi}}(1-2\nu^*) \cdot L_2 n^{(1-r)-q}, & q > (1-r) \end{cases}, \tag{3.57a}$$

$$\mathsf{SU}_b^U(n) := \begin{cases} \sqrt{\frac{2}{\pi}}(1-2\nu^*) \left(1 + U_1 n^{q-(1-r)}\right)^{-1}, & q < (1-r) \\ \sqrt{\frac{2}{\pi}}(1-2\nu^*) \cdot U_2 n^{(1-r)-q}, & q > (1-r) \end{cases}. \tag{3.57b}$$

Proof. Note that Equations (3.53) and (3.54) imply that the conditions $r_s(\mathbf{\Sigma}) \geq bn$ and $r_s(\mathbf{\Sigma}_{-\tau}) \geq b_2 n$ are clearly satisfied for large enough n. Thus, we can apply Equation (3.29a) of Theorem 3 setting $k = s$ to get

$$\mathsf{SU}_b(1;n) \geq \sqrt{\frac{2}{\pi}}(1-2\nu^*)\frac{\lambda_1\left(\frac{(n-s)}{c\widetilde{\lambda}_{s+1}r_s(\mathbf{\Sigma}_{-1})} - \frac{c_3 n^{3/4}}{\lambda_{s+1}r_s(\mathbf{\Sigma})}\right)}{1+\lambda_1\left(\frac{cn}{\widetilde{\lambda}_{s+1}r_s(\mathbf{\Sigma}_{-1})} + \frac{c_4 n^{3/4}}{\lambda_{s+1}r_s(\mathbf{\Sigma})}\right)}$$

with probability at least $(1-5e^{-\sqrt{n}})$ over the training data. Substituting $s = n^r$ and $a = n^{-q}$, note that

$$\frac{\lambda_{s+1}r_s(\mathbf{\Sigma})}{\lambda_1} = \frac{\widetilde{\lambda}_{s+1}r_s(\mathbf{\Sigma}_{-1})}{\lambda_1} \asymp \frac{\frac{(1-\gamma)d}{d-s}n^p}{\frac{\gamma d}{s}} \asymp \frac{n^{p+r}}{n^{p-q}} \asymp n^{q+r}.$$

Substituting this above yields

$$\begin{aligned}\mathsf{SU}_b(1;n) &\geq \sqrt{\frac{2}{\pi}}(1-2\nu^*)\left(\frac{\frac{(n-n^r)}{cn^{q+r}} - \frac{c_3 n^{3/4}}{n^{q+r}}}{1+\frac{cn}{n^{q+r}} + \frac{c_4 n^{3/4}}{n^{q+r}}}\right) \\ &= \sqrt{\frac{2}{\pi}}(1-2\nu^*)\left(\frac{\frac{1}{c}\cdot\left(n^{(1-r)-q} - n^{-q}\right) - c_3\cdot n^{(3/4-r)-q}}{1+cn^{(1-r)-q} + c_4\cdot n^{(3/4-r)-q}}\right).\end{aligned}$$

Thus, there are two cases:

1. $0 < q \leq (1-r)$, in which case the terms corresponding to $n^{q-(1-r)}$ dominate, and there exists universal constant L_1 such that

$$\mathsf{SU}_b(1;n) \geq \sqrt{\frac{2}{\pi}}(1-2\nu^*)\left(1+L_1 n^{q-(1-r)}\right)^{-1}.$$

2. $q > (1-r)$, in which case the numerator goes to 0 but the denominator goes to 1 as $n \to \infty$, and so there exists universal constant L_2 such that

$$\mathsf{SU}_b(1;n) \geq \sqrt{\frac{2}{\pi}}(1-2\nu^*)\cdot L_2 n^{(1-r)-q}.$$

This completes the proof of the lower bound. An almost identical argument gives the proof of the upper bound, so we omit it here. □

Observe that for $q > (1-r)$, the true signal does not survive at all, i.e. $\mathsf{SU}_r(1;n) \to 0$ as $n \to \infty$. Interestingly, for $q \leq (1-r)$, there is also non-trivial attenuation of signal when binary labels are interpolated, i.e. $\mathsf{SU}_r(1;n) \to \sqrt{\frac{2}{\pi}}\cdot(1-2\nu^*) < 1$ as $n \to \infty$. At a high level, this is a consequence of effective misspecification induced by the sign operator on real

output. As mentioned in the discussion in Section 3.4, this is also spiritually related to the attenuation factor of signal that has been traditionally been observed as a result of 1-bit quantization applied to a matched filter [142, 24].

As we will see in the following lemma, the corresponding case leads to zero attenuation of signal when real output is interpolated., i.e. $\mathsf{SU}_r(1;n) \to 1$.

Lemma 10 (Survival for interpolation of real output)**.** *There exist universal positive constants* $(L_1, U_1, L_2, U_2, \overline{L_1}, \overline{U_1}, \overline{L_2}, \overline{U_2})$ *such that for sufficiently large* n*, we have*

$$\mathsf{SU}_r^L(n) \leq \mathsf{SU}_r(1;n) \leq \mathsf{SU}_r^U(n),$$

with probability at least $(1 - 8e^{-\sqrt{n}})$ *over the randomness in the training data* $\{X_i, Y_i\}_{i=1}^n$*, where we denote*

$$\mathsf{SU}_r^L(n) := \begin{cases} \left(1 + L_1 n^{q-(1-r)}\right)^{-1}, & q < (1-r) \\ L_2 n^{(1-r)-q}, & q > (1-r) \end{cases}, \tag{3.58a}$$

$$\mathsf{SU}_r^U(n) := \begin{cases} \left(1 + U_1 n^{q-(1-r)}\right)^{-1}, & q < (1-r) \\ U_2 n^{(1-r)-q}, & q > (1-r) \end{cases}. \tag{3.58b}$$

Equivalently, we can write

$$\overline{\mathsf{SU}_r}^L(n) \leq 1 - \mathsf{SU}_r(1;n) \leq \overline{\mathsf{SU}_r}^U(n),$$

where we denote

$$\overline{\mathsf{SU}_r}^L(n) := \begin{cases} \overline{L_1} n^{q-(1-r)}, & q < (1-r) \\ \left(1 + \overline{L_2} n^{(1-r)-q}\right)^{-1}, & q > (1-r) \end{cases}, \tag{3.59a}$$

$$\overline{\mathsf{SU}_r}^U(n) := \begin{cases} \overline{U_1} n^{q-(1-r)}, & q < (1-r) \\ \left(1 + \overline{U_2} n^{(1-r)-q}\right)^{-1}, & q > (1-r) \end{cases} \tag{3.59b}$$

Proof. The proof follows by substituting the spectrum of the bi-level covariance model into the upper and lower bounds of survival from Equations (3.30b) and (3.30a). This is essentially an identical argument to the proof of Lemma 9, and so we omit it here. □

Observe that for the case of interpolation of real output, we have additionally computed bounds on the quantity $(1-\mathsf{SU}_r(1;n))$, which will subsequently be useful for the computation of bounds on contamination. We have not stated this here to avoid complicating the proof, but it is interesting to note that if the real-valued output had a non-zero level of independent additive zero-mean Gaussian noise, then this would not matter for the scaling of the survival results asymptotically — this is a consequence of the range of parameter choices that we have chosen for our bi-level ensemble. Such label noise would effectively be completely absorbed by the excess features.

We now state an upper bound on contamination for the bi-level ensemble.

Lemma 11 (Contamination for interpolation of binary labels)**.** *There are universal positive constants (U_3, U_4 and U_5) such that for large enough n, we have $\mathsf{CN}_b(1;n) \leq \mathsf{CN}_b^U(n)$ with probability at least $\left(1-\frac{4}{n}\right)$ over the randomness in the training data $\{X_i, Y_i\}_{i=1}^n$, where we denote*

$$\mathsf{CN}_b^U(n) = \begin{cases} U_3 n^{\frac{-\min\{(p-1),(1-r)\}}{2}} \cdot \sqrt{\ln n} \text{ if } q < (1-r) \\ U_4 n^{\frac{-\min\{(p-1),(2q+r-1)\}}{2}} \cdot \sqrt{\ln n} \text{ if } q > (1-r) \end{cases} \tag{3.60}$$

Proof. We start by proving the statement for the case $q \leq (1-r)$. From Equations (3.53) and (3.54), we showed that for large enough n, we have $r_s(\boldsymbol{\Sigma}_{-1}) \asymp n^p \gg n$. Substituting $k = l = s$ in Equation (3.31a) from Theorem 4, we have

$$\mathsf{CN}_b(1;n) \leq c_7 \cdot \sqrt{\left(\frac{s}{n} + n \cdot \frac{\sum_{j>s} \widetilde{\lambda}_j^2}{\left(\sum_{j>s} \widetilde{\lambda}_j\right)^2}\right) \cdot \ln n \cdot (1 + \mathsf{SU}_b(1;n)^2)} \tag{3.61}$$

almost surely for every realization of SU with probability at least $\left(1-\frac{3}{n}\right)$ over the training data. We first evaluate the term

$$T_1 := \frac{s}{n} + n \cdot \frac{\sum_{j>s} \widetilde{\lambda}_j^2}{\left(\sum_{j>s} \widetilde{\lambda}_j\right)^2}.$$

First, note that

$$\sum_{j>s} \widetilde{\lambda}_j^2 = (d-s-1)\left(\frac{(1-\gamma)d}{d-s}\right)^2 \asymp d = n^p \text{ and}$$
$$\left(\sum_{j>s} \widetilde{\lambda}_j\right)^2 = \left((d-s-1)\frac{(1-\gamma)d}{d-s}\right)^2 \asymp n^{2p}.$$

Using this, we obtain

$$T_1 \asymp n^{(r-1)} + n^{(1-p)} \asymp n^{-\min\{(p-1),(1-r)\}}. \tag{3.62}$$

Now, from Equation (3.57b), we get (for large enough n)

$$\mathsf{SU}_b(1;n) \leq \mathbb{1}_{q\leq(1-r)}\sqrt{\frac{2}{\pi}}\left(1 + U_1 n^{q-(1-r)}\right)^{-1} + \mathbb{1}_{q>(1-r)} U_2 n^{(1-r)-q} \leq \max\left\{U_2, \sqrt{\frac{2}{\pi}}\right\} \tag{3.63}$$

with probability at least $(1 - 4e^{-p_1 n})$ over the training data. Substituting Equations (3.62) and (3.63) in Equation (3.61), we have

$$\mathsf{CN}_b(1;n) \leq U_3 n^{-\frac{\min\{(p-1),(1-r)\}}{2}} \cdot \sqrt{\ln n}$$

with probability at least $\left(1 - \frac{4}{n}\right)$ for appropriately defined positive constant U_3. This completes the proof for the first case.
Now, we move on to the second case, i.e. $q > (1 - r)$. From Equations (3.55) and (3.56), we saw that in this case, we have $r_0(\mathbf{\Sigma}_{-1}) \asymp n^{q+r} \gg n$. Substituting $k = l = 0$ in Equation (3.31a) from Theorem 4, we have

$$\mathsf{CN}_b(1; n) \leq c_7 \cdot \sqrt{\left(n \cdot \frac{\sum_{j>0} \widetilde{\lambda}_j^2}{\left(\sum_{j>0} \widetilde{\lambda}_j\right)^2} \right) \cdot \ln n \cdot (1 + \mathsf{SU}_b(1; n)^2)}$$

with probability at least $\left(1 - \frac{3}{n}\right)$ over the training data. As before, we evaluate the term

$$T_1 := n \cdot \frac{\sum_{j>0} \widetilde{\lambda}_j^2}{(\sum_{j>0} \widetilde{\lambda}_j)^2}$$

By a calculation very similar to the one in Appendix C.2 of [104], we get

$$\sum_{j>0} \widetilde{\lambda}_j^2 = (s-1) \cdot \frac{a^2 d^2}{s^2} + (d - s - 1) \cdot \frac{(1-a)^2 d^2}{(d-s)^2} \asymp n^{2p+2q-r} + n^p.$$

Moreover, we get $(\sum_{j>0} \widetilde{\lambda}_j)^2 = (d - \frac{ad}{s})^2 = (n^p - n^{p-(r+q)})^2 \asymp n^{2p}$ since $(q+r) > 0$. Therefore, we get

$$T_1 \asymp n^{(1-p)} + n^{(1+2q-r)} \asymp n^{-\min\{(p-1),(2q+r-1)\}}.$$

The other steps proceed as for the first case, and substituting this expression for the term T_1 completes the proof for the second case. □

For some parameterizations of the bi-level ensemble, we can get a slightly more sophisticated upper bound on contamination when the labels interpolated are real, as detailed in the following lemma.

Lemma 12 (Contamination for interpolation of real output)**.** *For universal positive constants (U_3, U_4, U_5) and large enough n, we have $\mathsf{CN}_r(1; n) \leq \mathsf{CN}_r^U(n)$ with probability at least $\left(1 - \frac{3}{n}\right)$ over the randomness in the training data $\{X_i, Y_i\}_{i=1}^n$, where we denote*

$$\mathsf{CN}_r^U(n) = \begin{cases} U_3 n^{q-(1-r)-\frac{\min\{(p-1),(1-r)\}}{2}} \cdot \sqrt{\ln n}, & q < (1-r), \\ U_4 n^{-\frac{\min\{(p-1),(2q+r-1)\}}{2}} \cdot \sqrt{\ln n}, & q > (1-r) \end{cases}. \tag{3.64}$$

Proof. We follow an identical approach as in the proof of Lemma 11 to bound the term T_1. Substituting this along with the upper bound on the quantity $(1 - \mathsf{SU}_r(1; n))$ from Equation (3.59b) (Lemma 10) in Equation (3.32a), and using the fact that $\mathsf{SU}_r(1; n) \leq 1$, Equation (3.64) follows for appropriately defined positive constants (U_3, U_4). This completes the proof. □

Finally, we state and prove our lower bounds on contamination together for interpolation of binary labels as well as real output.

Lemma 13 (Lower bounds on contamination)**.** *There are universal positive constants* (L_3, L_4, p_2) *such that for large enough* n*, we have* $\mathsf{CN}_b(1;n), \mathsf{CN}_r(1;n) \geq \mathsf{CN}^L(n)$ *with probability at least* $(1-2e^{-p_2 n})$ *over the randomness in the training data* $\{X_i, Y_i\}_{i=1}^n$*, where we define*

$$\mathsf{CN}^L(n) := \begin{cases} L_3 n^{q-(1-r)-\frac{p-1}{2}}, & q < (1-r) \\ L_4 n^{-\frac{(p-1)}{2}}, & q > (1-r) \end{cases}. \tag{3.65}$$

Proof. Using Equation (3.54) we have, for large enough n, $r_s\left(\mathbf{\Sigma}_{-1}^2\right) \asymp n^p \gg n$. Taking $k = s$ in Equation (3.31b) from Theorem 4, for universal constants c_8, c_9, with probability at least $(1-2e^{-\frac{n}{c_8}})$, we have

$$\begin{aligned} \mathsf{CN}_b(1;n) &\geq \sqrt{n}.\frac{\sqrt{r_s\left(\mathbf{\Sigma}_{-1}^2\right)\widetilde{\lambda}_{s+1}^2}}{c_9\left(\sum_{j=1}^{d}\lambda_j + \lambda_1 n\right)} \\ &\asymp n^{\frac{1}{2}} \cdot \frac{\sqrt{n^p\left(\frac{(1-\gamma)d}{d-s}\right)^2}}{d + n\frac{\gamma d}{s}} \\ &\asymp \frac{n^{-\frac{(p-1)}{2}}}{1+n^{(1-r)-q}}, \\ &\asymp \begin{cases} n^{q-(1-r)-\frac{(p-1)}{2}}, & q < (1-r) \\ n^{-\frac{(p-1)}{2}}, & q > (1-r) \end{cases}. \end{aligned}$$

Thus Equation (3.60) follows by choosing appropriate constants p_2, L_3 and L_4, completing the proof.

□

Comparing the upper bound (Equation (3.64)) and lower bound (Equation (3.65)) for the case of interpolating real output, we observe that these bounds would be matching up to constant factors *iff* $(p-1) \leq (1-r)$. In addition to the above condition, the upper bound for interpolation of binary labels (Equation (3.61)) will match the lower bound *iff* $q > (1-r)$.

Finally, we compute bounds on the ratio of survival to contamination, $\mathsf{SU}_b(1;n)/\mathsf{CN}_b(1;n)$, for the interpolation of binary labels. A directly substitution of the upper and lower bounds for $\mathsf{SU}_b(1;n)$ and $\mathsf{CN}_b(1;n)$ from Equations (3.57a), (3.57b) in Lemma 9, Equations (3.60) in Lemma 11 and Equation (3.65) in Lemma 13, gives us (for large enough n)

$$\mathsf{SNR}^L(n) \leq \frac{\mathsf{SU}_b(1;n)}{\mathsf{CN}_b(1;n)} \leq \mathsf{SNR}^U(n), \tag{3.66}$$

with probability at least $\left(1-\frac{16}{n}\right)$ over the training data, where we denote

$$\mathsf{SNR}^L(n) := \begin{cases} L_5 \cdot n^{\frac{\min\{(p-1),(1-r)\}}{2}} \cdot (\ln n)^{-\frac{1}{2}}, & 0 < q < (1-r) \\ L_6 \cdot n^{\frac{\min\{(p-1),(2q+r-1)\}}{2}+(1-r)-q} \cdot (\ln n)^{-\frac{1}{2}}, & q > (1-r) \end{cases}. \tag{3.67a}$$

$$\mathsf{SNR}^U(n) = U_5 \cdot n^{\frac{p-1}{2}+(1-r)-q}. \tag{3.67b}$$

Proof of Theorem 2

We are now ready to complete the proof of Theorem 2. First we compute a lower bound on regression test loss. From Equations (3.23), (3.59a) and (3.65), we have (for large enough n)

$$\begin{aligned} \mathcal{E}_{\text{reg}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}; n) &= (1-\mathsf{SU}_r(1;n))^2 + (\mathsf{CN}_r(1;n))^2 \\ &\geq (\overline{\mathsf{SU}_r}^L(n))^2 + (\mathsf{CN}_r^L(n))^2 \\ &= \begin{cases} \overline{L_1}^2 n^{2(q-(1-r))} + L_3^2 n^{-2(1-r)-(p-1)+2q}, & q < (1-r) \\ \left(1+\overline{L_2} n^{(1-r)-q}\right)^{-2} + L_4^2 n^{-(p-1)}, & q > (1-r) \end{cases} \end{aligned}$$

with probability at least $(1-2e^{-\sqrt{n}}-2e^{-p_2 n})$. Thus, we have

$$\liminf_{n\to\infty} \mathcal{E}_{\text{reg}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}; n) \geq \begin{cases} 0, & q < (1-r) \\ 1, & q > (1-r) \end{cases}$$

with probability equal to 1. Next, we compute an upper bound on regression test loss. From Equations (3.23), (3.59b) and (3.64), we have (for large enough n)

$$\begin{aligned} \mathcal{E}_{\text{reg}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}; n) &\leq (\overline{\mathsf{SU}_r}^U(n))^2 + (\mathsf{CN}_r^U(n))^2 \\ &= \begin{cases} \overline{U_1}^2 n^{2(q-(1-r))} + U_3^2 n^{-2(1-r)-\min\{(p-1),(1-r)\}+2q} \ln n, & q < (1-r) \\ \left(1+\overline{U_2} n^{(1-r)-q}\right)^{-2} + U_4^2 n^{-(p-1)} \ln n, & q > (1-r) \end{cases} \end{aligned}$$

with probability at least $\left(1-2e^{-\sqrt{n}}-\frac{3}{n}\right)$. Thus, we have

$$\limsup_n \mathcal{E}_{\text{reg}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}; n) \leq \begin{cases} 0, & q < (1-r) \\ 1, & q > (1-r) \end{cases}$$

with probability equal to 1. By the sandwich theorem, we get

$$\lim_{n\to\infty} \mathcal{E}_{\text{reg}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}; n) = \begin{cases} 0, & q < (1-r) \\ 1, & q > (1-r) \end{cases}$$

with probability 1, completing our characterization of regression.

We now move on to our final characterization of binary classification test loss, starting with the upper bound. By Proposition 1, we have

$$\mathcal{E}_{\text{binary}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}};n)=\frac{1}{2}-\frac{1}{\pi}\mathsf{tan}^{-1}\left(\frac{\mathsf{SU}_b(1;n)}{\mathsf{CN}_b(1;n)}\right).$$

From Equation (3.66), we get

$$\frac{1}{2}-\frac{1}{\pi}\mathsf{tan}^{-1}\left(\mathsf{SNR}^U(n)\right)\leq\mathcal{E}_{\text{binary}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}};n)\leq\frac{1}{2}-\frac{1}{\pi}\mathsf{tan}^{-1}\left(\mathsf{SNR}^L(n)\right).$$

Taking the limit as $n\to\infty$ in Equation (3.67a), we have

$$\liminf_{n\to\infty}\mathsf{SNR}^L(n)=\begin{cases}\infty, & q<\frac{\min\{(p-1),(2q+r-1)\}}{2}+(1-r)\\ 0, & q>\frac{\min\{(p-1),(2q+r-1)\}}{2}+(1-r)\end{cases}.$$

with probability 1. Thus, we have

$$\limsup_n\mathcal{E}_{\text{binary}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}};n)\leq\begin{cases}0, & q<\frac{\min\{(p-1),(2q+r-1)\}}{2}+(1-r)\\ \frac{1}{2}, & q>\frac{\min\{(p-1),(2q+r-1)\}}{2}+(1-r)\end{cases}.$$

with probability 1. To simplify further, consider the case for which $(2q+r-1)<(p-1)$. Then, the condition becomes $q<q+\frac{(r-1)}{2}+(1-r)=\frac{(1-r)}{2}\implies\frac{(1-r)}{2}>0$, which is always true under the bi-level ensemble (as $r<1$). Thus, we can effectively ignore this argument, and simply write

$$\limsup_n\mathcal{E}_{\text{binary}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}};n)\leq\begin{cases}0, & q<\frac{(p-1)}{2}+(1-r)\\ \frac{1}{2}, & q>\frac{(p-1)}{2}+(1-r)\end{cases}.$$

On the other hand, we can also compute the limiting upper bound on SNR:

$$\limsup_n\mathsf{SNR}^U(n)=\begin{cases}\infty, & 0<q<\frac{(p-1)}{2}+(1-r)\\ 0, & q>\frac{(p-1)}{2}+(1-r).\end{cases}$$

and so the binary classification test loss is *lower bounded* by:

$$\liminf_n\mathcal{E}_{\text{binary}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}};n)\geq\begin{cases}0, & 0<q<\frac{(p-1)}{2}+(1-r)\\ \frac{1}{2}, & q>\frac{(p-1)}{2}+(1-r).\end{cases}$$

Putting these together, we get

$$\lim_{n\to\infty}\mathcal{E}_{\text{binary}}(\widehat{\boldsymbol{\alpha}}_{2,\mathsf{binary}};n)=\begin{cases}0, & 0<q<\frac{(p-1)}{2}+(1-r)\\ \frac{1}{2}, & q>\frac{(p-1)}{2}+(1-r).\end{cases}$$

This completes the proof. □

3.8 Appendix: Technical lemmas

Proof of Lemma 4

In this subsection, we prove Lemma 4, i.e. equivalent quadratic form expressions for the contamination factor when binary labels are interpolated. As argued in Section 3.6, for any $j \in \{1, \dots, d\}$, the coefficient $\widehat{\boldsymbol{\alpha}}_j$ is given by

$$\widehat{\boldsymbol{\alpha}}_j = \mathbf{e}_j^\top \boldsymbol{\Phi}_{\mathsf{train}} \mathbf{A}^{-1} \mathbf{Y}_{\mathsf{train}} = \sqrt{\lambda_j} \mathbf{z}_j^\top \mathbf{A}^{-1} \mathbf{y}_\tau.$$

From the Sherman-Morrison-Woodbury identity, we have

$$\mathbf{A}^{-1} = \mathbf{A}_{-\tau}^{-1} - \frac{\lambda_\tau \mathbf{A}_{-\tau}^{-1} \mathbf{z}_\tau \mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1}}{1 + \lambda_\tau \mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1} \mathbf{z}_\tau}.$$

Using this, we can rewrite $\widehat{\boldsymbol{\alpha}}_j$ as

$$\begin{aligned}
\widehat{\boldsymbol{\alpha}}_j &= \sqrt{\lambda_j} \mathbf{z}_j^\top \left(\mathbf{A}_{-\tau}^{-1} - \frac{\lambda_\tau \mathbf{A}_{-\tau}^{-1} \mathbf{z}_\tau \mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1}}{1 + \lambda_\tau \mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1} \mathbf{z}_\tau} \right) \mathbf{y}_\tau \\
&= \sqrt{\lambda_j} \cdot \left(1 - \frac{1}{1 + \lambda_\tau \mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1} \mathbf{z}_\tau} \right) \cdot \mathbf{z}_j^\top \mathbf{A}_{-\tau}^{-1} \mathbf{y}_\tau \\
&= \sqrt{\lambda_j} \cdot \mathbf{z}_j^\top \mathbf{A}_{-\tau}^{-1} \left(\mathbf{y}_\tau - \mathsf{SU}_b(\tau) \mathbf{z}_\tau \right)
\end{aligned}$$

where the last equality follows from Equation (3.43).
Using the definition of contamination (Equation (3.22)) and the above expressions, we get

$$\begin{aligned}
\mathsf{CN}_b^2(\tau) = \sum_{j=1, j \neq \tau}^{d} \lambda_j \widehat{\boldsymbol{\alpha}}_j^2 &= \sum_{j=1, j \neq \tau}^{d} \lambda_j^2 \mathbf{y}_\tau^\top \mathbf{A}^{-1} \mathbf{z}_j \mathbf{z}_j^\top \mathbf{A}^{-1} \mathbf{y}_\tau \\
&= \mathbf{y}_\tau^\top \mathbf{A}^{-1} \left(\sum_{j=1, j \neq \tau}^{d} \lambda_j^2 \mathbf{z}_j \mathbf{z}_j^\top \right) \mathbf{A}^{-1} \mathbf{y}_\tau \\
&= \mathbf{y}_\tau^\top \mathbf{C} \mathbf{y}_\tau.
\end{aligned}$$

Now, we denote $\widetilde{\mathbf{y}_\tau} := \mathbf{y}_\tau - \mathsf{SU}_b(\tau) \mathbf{z}_\tau$. To prove the second form of contamination, we use the following sequence of equalities:

$$\begin{aligned}
\mathsf{CN}_b^2(\tau) = \sum_{j=1, j \neq \tau}^{d} \lambda_j \widehat{\boldsymbol{\alpha}}_j^2 &= \sum_{j=1, j \neq \tau}^{d} \lambda_j \left(\sqrt{\lambda_j} \mathbf{z}_j^\top \mathbf{A}_{-\tau}^{-1} \widetilde{\mathbf{y}_\tau} \right)^2 \\
&= \sum_{j=1, j \neq \tau}^{d} \lambda_j^2 \widetilde{\mathbf{y}_\tau}^\top \mathbf{A}_{-\tau}^{-1} \mathbf{z}_j \mathbf{z}_j^\top \mathbf{A}_{-\tau}^{-1} \widetilde{\mathbf{y}_\tau} \\
&= \widetilde{\mathbf{y}_\tau}^\top \mathbf{A}_{-\tau}^{-1} \left(\sum_{j=1, j \neq \tau}^{d} \lambda_j^2 \mathbf{z}_j \mathbf{z}_j^\top \right) \mathbf{A}_{-\tau}^{-1} \widetilde{\mathbf{y}_\tau} \\
&= \widetilde{\mathbf{y}_\tau}^\top \widetilde{\mathbf{C}} \widetilde{\mathbf{y}_\tau}.
\end{aligned}$$

This completes the proof of Lemma 4. □

Proof of Lemma 5

In this subsection, we prove Lemma 5, i.e. a high-probability upper bound on the quadratic forms $\widetilde{\mathbf{y}_\tau}^\top \widetilde{\mathbf{C}} \widetilde{\mathbf{y}_\tau}$ and $\mathbf{z}_\tau^\top \widetilde{\mathbf{C}} \mathbf{z}_\tau$ over only the randomness in $\{\mathbf{z}_\tau, \mathbf{y}_\tau\}$. Recall that we defined the random variables $\{\mathbf{z}_\tau, \mathbf{y}_\tau\}$ in Section 3.6. Note that $\widetilde{\mathbf{C}}$ is almost surely positive definite and $\{\mathbf{z}_\tau, \widetilde{\mathbf{y}_\tau}\}$ are both pairwise independent of $\widetilde{\mathbf{C}}$. Further, note that

$$\begin{aligned}
\widetilde{\mathbf{y}_\tau}^\top \widetilde{\mathbf{C}} \widetilde{\mathbf{y}_\tau} &= (\mathbf{y}_\tau - \mathsf{SU}_b(\tau)\mathbf{z}_\tau)^\top \widetilde{\mathbf{C}} (\mathbf{y}_\tau - \mathsf{SU}_b(\tau)\mathbf{z}_\tau) \\
&\leq (\mathbf{y}_\tau - \mathsf{SU}_b(\tau)\mathbf{z}_\tau)^\top \widetilde{\mathbf{C}} (\mathbf{y}_\tau - \mathsf{SU}_b(\tau)\mathbf{z}_\tau) + (\mathbf{y}_\tau + \mathsf{SU}_b(\tau)\mathbf{z}_\tau)^\top \widetilde{\mathbf{C}} (\mathbf{y}_\tau + \mathsf{SU}_b(\tau)\mathbf{z}_\tau) \\
&= 2\mathbf{y}_\tau^\top \widetilde{\mathbf{C}} \mathbf{y}_\tau + 2\mathsf{SU}_b(\tau)^2 \mathbf{z}_\tau^\top \widetilde{\mathbf{C}} \mathbf{z}_\tau.
\end{aligned}$$

From Equation (3.40), we have

$$\mathbf{z}_\tau^\top \widetilde{\mathbf{C}} \mathbf{z}_\tau \leq \mathsf{tr}(\widetilde{\mathbf{C}}) \left(1 + \frac{1}{c}\right) \cdot (\ln n)$$

almost surely for every realization of the random matrix $\widetilde{\mathbf{C}}$, and with probability at least $\left(1 - \frac{1}{n}\right)$ over the randomness in $\mathbf{z}_\tau$. By an identical argument (noting that $y_{\tau,i}^2 = 1$ almost surely, and that $\mathbb{E}\left[y_{\tau,i} y_{\tau,j}\right] = 0$ for any $i \neq j$), we can show that

$$\mathbf{z}_\tau^\top \widetilde{\mathbf{C}} \mathbf{z}_\tau \leq \mathsf{tr}(\widetilde{\mathbf{C}}) \left(1 + \frac{1}{c}\right) \cdot (\ln n) \tag{3.68}$$

Substituting these inequalities in the expression for $\widetilde{\mathbf{y}_\tau}^\top \widetilde{\mathbf{C}} \widetilde{\mathbf{y}_\tau}$ completes the proof. □

Proof of Lemma 7

In this subsection, we prove Lemma 7, i.e. a high-probability lower bound on the minimum eigenvalue of the random (almost surely positive semidefinite) matrix $\mathbf{C}$. Recall that we defined

$$\begin{aligned}
\mathbf{C} &:= \mathbf{A}^{-1} \left(\sum_{j=1, j \neq \tau}^{d} \lambda_j^2 \mathbf{z}_j \mathbf{z}_j^\top \right) \mathbf{A}^{-1}, \\
&= \mathbf{A}^{-1} \left(\sum_{j=1}^{d-1} \widetilde{\lambda}_j^2 \mathbf{z}_j \mathbf{z}_j^\top \right) \mathbf{A}^{-1}.
\end{aligned}$$

Using the mathematical fact from Section 3.9, we have

$$\mu_n(\mathbf{C}) \geq (\mu_n(\mathbf{A}^{-1}))^2 \mu_n \left(\sum_{j=1}^{d-1} \widetilde{\lambda}_j^2 \mathbf{z}_j \mathbf{z}_j^\top \right).$$

Now, Equations (3.37) and (3.38a) from Lemma 1 can be used to lower bound the terms $(\mu_n(\mathbf{A}^{-1}))^2$ and $\mu_n\left(\sum_{j=1}^{d-1}\widetilde{\lambda}_j^2\mathbf{z}_j\mathbf{z}_j^\top\right)$ respectively. Substituting these lower bounds into the above bound completes the proof. □

Proof of Lemma 8

In this subsection, we prove Lemma 8, i.e. equivalent quadratic form expressions for the contamination factor when real output is interpolated. This proof closely mirrors the proof of Lemma 4.

Let $\widehat{\boldsymbol{\alpha}}_j$ denote the j^{th} component of $\widehat{\boldsymbol{\alpha}}_{2,\mathsf{real}}$. As argued in Section 3.6, for any $j \in \{1,\ldots,d\}$, the coefficient $\widehat{\boldsymbol{\alpha}}_j$ is given by

$$\widehat{\boldsymbol{\alpha}}_j = \mathbf{e}_j^\top \boldsymbol{\Phi}_{\mathsf{train}}\mathbf{A}^{-1}\mathbf{Z}_{\mathsf{train}} = \sqrt{\lambda_j}\mathbf{z}_j^\top\mathbf{A}^{-1}\mathbf{z}_\tau. \tag{3.69}$$

By the Sherman-Morrison-Woodbury Identity, we have

$$\mathbf{A}^{-1} = \mathbf{A}_{-\tau}^{-1} - \frac{\lambda_\tau\mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau\mathbf{z}_\tau^\top\mathbf{A}_{-\tau}^{-1}}{1+\lambda_\tau\mathbf{z}_\tau^\top\mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau}.$$

Using this, we can rewrite $\widehat{\boldsymbol{\alpha}}_j$ as

$$\begin{aligned}\widehat{\boldsymbol{\alpha}}_j &= \sqrt{\lambda_j}\left(1 - \frac{\lambda_\tau\mathbf{z}_\tau^\top\mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau}{1+\lambda_\tau\mathbf{z}_\tau^\top\mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau}\right)\mathbf{z}_j^\top\mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau \\ &= \sqrt{\lambda_j}(1-\mathsf{SU}_r(\tau))\mathbf{z}_j^\top\mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau,\end{aligned} \tag{3.70}$$

where the last equality follows from Equation (3.48).

Finally, using the definition of contamination (Equation (3.22)) together with Equation (3.69) gives us

$$\begin{aligned}\mathsf{CN}_r^2(\tau) = \sum_{j=1,j\neq\tau}^{d}\lambda_j\widehat{\boldsymbol{\alpha}}_j^2 &= \sum_{j=1,j\neq\tau}^{d}\lambda_j^2\mathbf{z}_\tau^\top\mathbf{A}^{-1}\mathbf{z}_j\mathbf{z}_j^\top\mathbf{A}^{-1}\mathbf{z}_\tau \\ &= \mathbf{z}_\tau^\top\mathbf{A}^{-1}\left(\sum_{j=1,j\neq\tau}^{d}\lambda_j^2\mathbf{z}_j\mathbf{z}_j^\top\right)\mathbf{A}^{-1}\mathbf{z}_\tau \\ &= \mathbf{z}_\tau^\top\mathbf{C}\mathbf{z}_\tau.\end{aligned}$$

Similarly, applying Equation (3.70) gives us

$$\begin{aligned}
\mathsf{CN}_r^2(\tau) = \sum_{j=1, j\neq\tau}^{d} \lambda_j \widehat{\boldsymbol{\alpha}}_j^2 &= \sum_{j=1, j\neq\tau}^{d} \lambda_j \left(\sqrt{\lambda_j}(1-\mathsf{SU}_r(\tau))\mathbf{z}_j^\top \mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau\right)^2 \\
&= (1-\mathsf{SU}_r(\tau))^2 \sum_{j=1, j\neq\tau}^{d} \lambda_j^2 \mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1}\mathbf{z}_j\mathbf{z}_j^\top \mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau \\
&= (1-\mathsf{SU}_r(\tau))^2 \mathbf{z}_\tau^\top \mathbf{A}_{-\tau}^{-1} \left(\sum_{j=1, j\neq\tau}^{d} \lambda_j^2 \mathbf{z}_j\mathbf{z}_j^\top\right) \mathbf{A}_{-\tau}^{-1}\mathbf{z}_\tau \\
&= (1-\mathsf{SU}_r(\tau))^2 \mathbf{z}_\tau^\top \widetilde{\mathbf{C}}\mathbf{z}_\tau.
\end{aligned}$$

This completes the proof. □

3.9 Appendix: Mathematical facts

Upper bound on maximum eigenvalue of product of positive definite matrices

Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$ be symmetric positive definite matrices and let $\mathbf{C} = \mathbf{AB}$. It is a well known fact that for positive definite matrix $\mathbf{M}$, $\mu_1(\mathbf{M}) = \|\mathbf{M}\|_2$, i.e the largest eigenvalue is the operator norm. Using this,

$$\mu_1(\mathbf{C}) = \|\mathbf{C}\|_2 = \|\mathbf{AB}\|_2 \leq \|\mathbf{A}\|_2\|\mathbf{B}\|_2 = \mu_1(\mathbf{A})\mu_1(\mathbf{B}),$$

where the inequality follows from the sub-multiplicativity of operator norm.

Lower bound on minimum eigenvalue of product of positive definite matrices

Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$ be symmetric positive definite matrices and let $\mathbf{C} = \mathbf{AB}$. Note that since inverses exist for positive definite matrices we can write,

$$\mu_n(\mathbf{C}) = \frac{1}{\mu_1(\mathbf{C}^{-1})} \geq \frac{1}{\mu_1(\mathbf{A}^{-1})\mu_1(\mathbf{B}^{-1})} = \mu_n(\mathbf{A})\mu_n(\mathbf{B}),$$

where the inequality follows by applying the upper bound for eigenvalue of product of two positive definite matrices from Section 3.9.

Chapter 4

Multiclass classification

In the last chapter, we analyzed binary classification and contrasted it to regression. However, most practical real-world applications like image classification, object detection and tracking for autonomous systems, and recommendation algorithms for movies, songs, etc. are multiclass classification problems. Contemporary machine learning systems have shown tremendous success at these problems by use of gigantic models with a vast number of parameters that are trained on enormous datasets with a huge number of classes. Can we theoretically analyze the generalization performance for multiclass classification?

In this chapter we perform an asymptotic analysis of the error of the minimum-norm interpolating classifier for the multiclass classification problem with Gaussian features. We consider a bi-level ensemble model where the number of features, classes, favored features, and the feature weights themselves all scale with the number of training points. Under this model, Theorem 5 provides sufficient conditions for good generalization in the form of a region in which as the number of training points increase, the number of classes grows slowly enough, the total number of features (i.e. level of overparameterization) grows fast enough, the number of favored features grows slowly enough, and the amount of favoring of those favored features is sufficient to allow for asymptotic generalization.

To prove our main result, Theorem 5, we present a novel typicality-style argument featuring the feature margin (gap between the largest and second-largest feature) for computing sufficient conditions for correct classification utilizing the signal-processing inspired concepts of survival and contamination from Chapters 2 and 3 and leveraging the random-matrix analysis tools sharpened in [10].

The key is analyzing what happens with multiclass training data where there are relatively fewer positive examples of each class, and where the training data for a particular class is not independent of the features corresponding to other classes. The analysis shows that as a result of having fewer positive exemplars for a class relative to the total size of the training data, the survival drops by a factor of k (the number of classes), while the contamination only drops by a factor of $\sqrt{k}$. As in binary classification, the ratio of the relevant survival to contamination terms plays the role of the effective signal-to-noise ratio and shows up as a key quantity in our error analysis (Equation (4.15) from Section 4.2). When this ratio grows

asymptotically to ∞, multiclass classification generalizes well.

4.1 Problem setup

We consider the multiclass classification problem with k classes. The training data consists of n pairs $\{\phi(X_i), \ell_i\}_{i=1}^n$ where $\phi(X_i) \in \mathbb{R}^d$ are i.i.d Gaussian vectors drawn from distribution[1]

$$\phi(X_i) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}).$$

Here, we assume without loss of generality (due to symmetry of the covariance matrix and the spectral theorem) that $\mathbf{\Sigma}$ is a diagonal matrix and its spectrum is given by $\boldsymbol{\lambda} := \begin{bmatrix}\lambda_1 & \dots & \lambda_d\end{bmatrix}$, where the eigenvalues are sorted in descending order, i.e. we have $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d > 0$.

We make the following assumption on how the labels $\ell_i \in [k]$ are generated.

Assumption 2. ***1-sparse orthogonal means model***

$$\ell_i = \underset{m \in [k]}{\operatorname{argmax}} \boldsymbol{\mu}_m^\top \phi(X_i). \tag{4.1}$$

Further we consider the simplified case where $\boldsymbol{\mu}_m$ are 1-sparse and orthogonal

$$\boldsymbol{\mu}_m = \frac{1}{\sqrt{\lambda_m}} \mathbf{e}_m,$$

where $\mathbf{e}_m$ is the unit vector with 1 at index m.

We define shorthand notation for the training data: let

$$\mathbf{\Phi}_{\mathsf{train}} := \begin{bmatrix}\phi(X_1) & \phi(X_2) & \cdots & \phi(X_n)\end{bmatrix}^\top \in \mathbb{R}^{n \times d}$$

denote the data (feature) matrix;

We use a one-hot encoding for representing the labels as the matrix $\mathbf{Y}^{oh} \in \mathbb{R}^{n \times k}$,

$$\mathbf{Y}^{oh} = \begin{bmatrix}\mathbf{y}_1^{oh} & \dots & \mathbf{y}_m^{oh} & \dots & \mathbf{y}_k^{oh}\end{bmatrix}, \tag{4.2}$$

where,

$$y_m^{oh}[i] = \begin{cases} 1, & \text{if } \ell_i = m \\ 0, & \text{otherwise} \end{cases}. \tag{4.3}$$

A zero-mean variant of the encoding where we subtract the mean $\frac{1}{k}$ from each entry is denoted:

$$\mathbf{y}_m = \mathbf{y}_m^{oh} - \frac{1}{k}\mathbf{1}. \tag{4.4}$$

[1] In the paper [133] an equivalent model where the Gaussian vectors have an identity covariance matrix but an explicit weighting is applied before performing interpolation is considered.

Our classifier consists of k coefficient vectors $\widehat{\boldsymbol{\alpha}}_m$ for $m \in [k]$ that are learned by minimum-norm interpolation of the zero-mean one-hot variants. [2]

$$\widehat{\boldsymbol{\alpha}}_m = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_2 \tag{4.5}$$

$$\text{s.t.} \quad \boldsymbol{\Phi}_{\mathsf{train}}\boldsymbol{\alpha} = \mathbf{y}_m. \tag{4.6}$$

We can express these coefficients in closed form as,

$$\widehat{\boldsymbol{\alpha}}_m = (\boldsymbol{\Phi}_{\mathsf{train}})^\top \left(\boldsymbol{\Phi}_{\mathsf{train}}(\boldsymbol{\Phi}_{\mathsf{train}})^\top\right)^{-1} \mathbf{y}_m. \tag{4.7}$$

On a test point $\phi(X) \sim \mathcal{N}(0, \boldsymbol{\Sigma})$ we predict the class label by computing k scalar "scores" and predict the class based on the largest score as follows:

$$\widehat{\ell} = \operatorname*{argmax}_{1\le m\le k} \widehat{\boldsymbol{\alpha}}_m^\top \phi(X). \tag{4.8}$$

The true label of the test point is $\ell_{test} = \operatorname{argmax}_{1\le m\le k} \boldsymbol{\mu}_m^\top \phi(X)$. A misclassification event $\mathcal{E}_{err}$ occurs iff

$$\operatorname*{argmax}_{1\le m\le k} \boldsymbol{\mu}_m^\top \phi(X) \neq \operatorname*{argmax}_{1\le m\le k} \widehat{\boldsymbol{\alpha}}_m^\top \phi(X).$$

In our work we determine sufficient conditions under which the probability of misclassification (computed over the randomness in both the training data and test point) goes to zero in an asymptotic regime where the number of training points, number of features, number of classes and feature weights scale according to the bi-level ensemble model similar to what we had in Definition 7. Note that this would imply that the more commonly computed quantity of test error, $\mathcal{E}_{\text{multi}}(\widehat{\boldsymbol{\alpha}}) = \mathbb{P}(\mathcal{E}_{\text{err}})$ computed only over the randomness in the test point, also goes to 0.

Definition 9 (Bi-level ensemble(n, p, q, r, t))**.** *The bi-level ensemble is parameterized by n, p, q, r, t, where*[3] *$p > 1, 0 \le r < 1$ and $0 < q < (p - r)$. Here, parameter p controls the extent of artificial overparameterization), r sets the number of preferred features, q controls the weights on preferred features and thus effective overparameterization, and t controls the number of classes. In particular, this ensemble sets parameters*

$$\begin{aligned} d &:= \lfloor n^p \rfloor \\ s &= \lfloor n^r \rfloor \\ a &= n^{-q} \text{ and} \\ k &= c_k \lfloor n^t \rfloor. \end{aligned} \tag{4.9}$$

[2] The classifier learned via this method is equivalent to those obtained by other natural training methods under sufficient overparameterization [146].

[3] We restrict (p, q, r) to this range to ensure that a) the regime is truly overparameterized (choice of p), b) the eigenvalues of the ensuing covariance matrix are always positive and ordered correctly (choice of q), c) the number of "high-energy" directions is sub-linear in n (choice of r).

The covariance matrix of the Gaussian features $\mathbf{\Sigma}(p, q, r)$ *is the diagonal matrix, whose entries are given by:*

$$\lambda_j = \begin{cases} \frac{ad}{s}, & 1 \leq j \leq s \\ \frac{(1-a)d}{d-s}, & \textit{otherwise.} \end{cases} \tag{4.10}$$

Note that under this bi-level ensemble model the true labels ℓ_i are simply generated as,

$$\ell_i = \underset{1 \leq m \leq k}{\operatorname{argmax}} \phi_(X_i)[m],$$

where we use the notation $\phi(X_i)[m]$ to refer to the m^{th} element of vector $\phi(X_i)$. Thus, a misclassfication event, $\mathcal{E}_{\text{err}}$, occurs iff,

$$\underset{1 \leq m \leq k}{\operatorname{argmax}} \phi(X)[m] \neq \underset{1 \leq m \leq k}{\operatorname{argmax}} \widehat{\boldsymbol{\alpha}}_m^\top \phi(X).$$

4.2 Main result

Theorem 5. ***(Asymptotic classification region in the bi-level model)****: Under the bi-level ensemble model 9, when the true data generating process is from a 1-sparse orthogonal means model (Assumption 2), the probability of misclassification* $\mathbb{P}(\mathcal{E}_{\text{err}}) \to 0$ *as* $n \to \infty$ *if the following conditions hold:*

$$\begin{aligned} t &< \min\left(r, 1-r, p+1-2(q+r), p-2, 2q+r-2\right) \\ q + r &> 1. \end{aligned}$$

Note that from Theorem 2 in Chapter 3, the condition $q+r > 1$ corresponds to the regime where the corresponding regression problem[4] does not generalize well and thus our result shows that multiclass classification can generalize in regimes where the regression problem does not. Figure 4.1 visualizes the regimes by considering slices of the four dimensional scaling parameter space of p, q, r and t. (1a) and (2a) fix the value of q to 0.75 and 0.95 respectively and contrast the multiclass problem with a fixed finite number of classes ($t = 0$) to the binary classification and regression problems. From these plots we observe that if we fix p, q, t and increase r, i.e. increasing how many features are favored (and thereby favoring each of them less), we transition from the regime where both regression and binary classification work, into the regime where binary classification works but regression does not, then the regime where this paper can prove multiclass classification works and finally to the regime where neither regression nor binary classification works.

In Figure 4.1, subplots (1b),(1c),(2b) and (2c) each visualize a slice along the r and t (class scaling) dimensions with fixed p and q. The x axis itself in these plots corresponds

[4]The corresponding regression problem is one where the true real number to be predicted is defined by a linear combination of favored features.

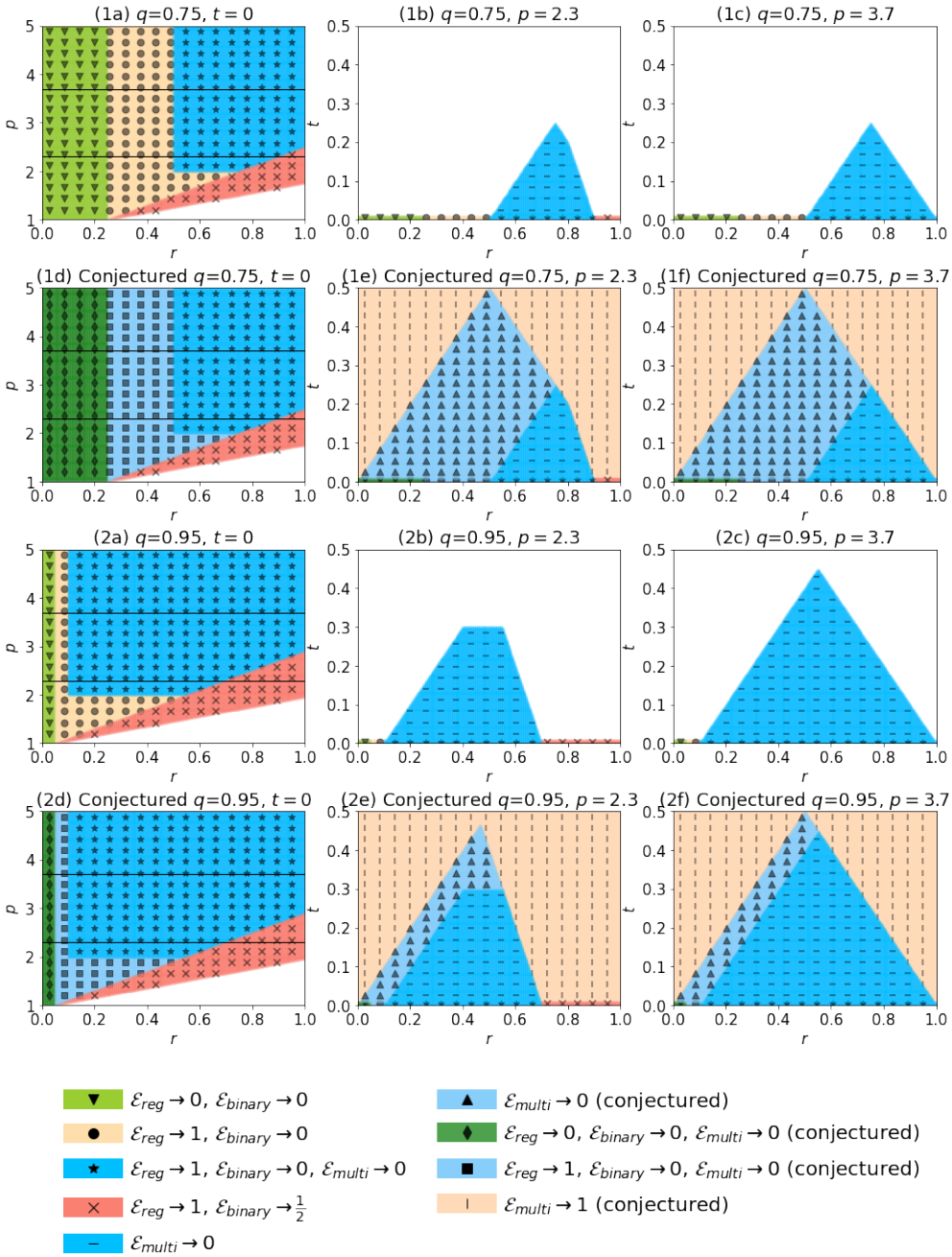


Figure 4.1. Visualization of the bi-level regimes in four dimensions p, q, r, t. (1a) and (2a) contrast multiclass classification with finite classes to binary classification and regression. The horizontal lines $p = 2.3$ and $p = 3.7$ correspond to the slices visualized in (1b), (1c), (2b) and (2c). The conjectured regimes are visualized in (1d), (1e), (1f), (2d), (2e) and (2f).

to a fixed finite classes setting. From (1b) we observe that the right-hand boundary of the region where multiclass classification generalizes well contains two slopes. These slopes arise

from the two conditions $t < 1 - r$ and $t < p + 1 - 2(q + r)$ in Theorem 5 and are a result of either contamination from favored (but not true) features dominating or contamination from the unfavored features dominating. In (1c) we are in the regime where binary classification works for all values of $r < 1$. However, as we increase t, eventually multiclass classification stops working.[5]

When we go from the binary problem to a multiclass problem with k classes, the survival drops by a factor of k as a consequence of having only $\frac{1}{k}$ fraction of positive training examples per class. This is because the one-hot labels we interpolate while training have fewer large values close to 1 that are able to positively correlate with the true feature vector. Having fewer positive exemplars also reduces the total energy in the training vector by a factor of k, and because of the square-root relationship of the standard deviation to the energy, the contamination only shrinks by a factor of $\sqrt{k}$. The overall survival/contamination ratio decreases by a factor of $\sqrt{k}$ making the multiclass classification task more difficult.[6] An interesting observation here is the amount of favoring required for good generalization is linked to the number of positive training examples per class. Indeed, if we consider a setting where the binary classification problem generalizes well, and we switch to the k class multiclass problem, then by increasing the number of training samples k fold (and thus matching the number of positive training examples per class in the multiclass case to the binary case) and keeping the number of features and feature weights constant we can generalize well for multiclass classification. (Section 4.5 elaborates on this phenomenon, as well as why it is somewhat surprising.)

Next, we present a brief overview of our proof that utilizes the survival/contamination analysis framework from Chapters 2 and 3 along with a typicality-inspired argument where the feature margin (difference between largest and second largest feature) on the test point plays a key role. The complete proof is provided in Sections 4.8, 4.9, 4.10, and 4.11.

Proof sketch

Assume without loss of generality that for the test point $\phi(X) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, the true class is τ for some $\tau \in [k]$.

[5]To be precise, what the region actually illustrates is that our proof approach stops being able to show that multiclass classification works. In the Conclusion section, we conjecture where we believe that multiclass classification actually stops working. The conjectured regions are illustrated in (1e),(1f),(2e) and (2f).

[6]This is also responsible for contamination due to favored features being able to cause errors. For binary classification, because the true feature survival is constant (depending only on the level of label noise), the survival can always asymptotically overcome any contamination from other favored features (See Section 3.7 from Chapter 3).

A necessary and sufficient condition for classification error is that for some $\zeta \neq \tau, \zeta \in [k]$,

$$\begin{aligned}\widehat{\alpha}_\tau[\tau]\phi(X)[\tau] + \widehat{\alpha}_\tau[\zeta]\phi(X)[\zeta] + \sum_{j \notin \{\tau,\zeta\}} \widehat{\alpha}_\tau[j]\phi(X)[j] < \widehat{\alpha}_\zeta[\tau]\phi(X)[\tau] \\ + \widehat{\alpha}_\zeta[\zeta]\phi(X)[\zeta] + \sum_{j \notin \{\tau,\zeta\}} \widehat{\alpha}_\zeta[j]\phi(X)[j] \\ \implies (\widehat{\alpha}_\tau[\tau] - \widehat{\alpha}_\zeta[\tau])\phi(X)[\tau] - (\widehat{\alpha}_\zeta[\zeta] - \widehat{\alpha}_\tau[\zeta])\phi(X)[\zeta] < \sum_{j \notin \{\tau,\zeta\}} (\widehat{\alpha}_\zeta[j] - \widehat{\alpha}_\tau[j])\phi(X)[j].\end{aligned}$$

Note that $\sum_{j \notin \tau,\zeta}$ refers to the sum over all feature indices 1 to d excluding τ and ζ.

Next note that $\phi(X) = \mathbf{\Sigma}^{\frac{1}{2}} Z$ for $Z \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ (Recall that $\mathbf{\Sigma}$ is a diagonal matrix with entries $\lambda_1, \lambda_2, \ldots, \lambda_d$). Thus, since τ is the true class for the test point X, under our 1-sparse orthogonal means model (Assumption 2) we have $Z[\tau] = \max_{1 \leq m \leq k} Z[m]$. The necessary and sufficient condition for error can then be rewritten as:

$$\lambda_\tau \widehat{h}_{\tau,\zeta}[\tau] Z[\tau] - \lambda_\zeta \widehat{h}_{\zeta,\tau}[\zeta] Z[\zeta] < \sum_{j \notin \{\tau,\zeta\}} \lambda_j \widehat{h}_{\zeta,\tau}[j] Z[j],$$

where we introduce the short-hand notation,

$$\widehat{h}_{\tau,\zeta}[j] = \lambda_j^{-1/2}(\widehat{\alpha}_\tau[j] - \widehat{\alpha}_\zeta[j]) \tag{4.11}$$

$$\widehat{h}_{\zeta,\tau}[j] = \lambda_j^{-1/2}(\widehat{\alpha}_\zeta[j] - \widehat{\alpha}_\tau[j])\text{..} \tag{4.12}$$

Since both τ and ζ are favored feature indices, by leveraging the definition of the bi-level model and denoting $\lambda_\tau = \lambda_\zeta = \lambda$, we get

$$\lambda \left(\widehat{h}_{\tau,\zeta}[\tau] Z[\tau] - \widehat{h}_{\zeta,\tau}[\zeta] Z[\zeta] \right) < \sum_{j \notin \{\tau,\zeta\}} \lambda_j \widehat{h}_{\zeta,\tau}[j] Z[j].$$

Next, we perform some algebraic manipulations,

$$\begin{aligned}\lambda \left(\widehat{h}_{\tau,\zeta}[\tau] Z[\tau] - \widehat{h}_{\zeta,\tau}[\zeta] Z[\zeta] \right) < \sum_{j \notin \{\tau,\zeta\}} \lambda_j \widehat{h}_{\zeta,\tau}[j] Z[j] \\ \implies \lambda \widehat{h}_{\tau,\zeta}[\tau](Z[\tau] - Z[\zeta]) + \lambda Z[\zeta](\widehat{h}_{\tau,\zeta}[\tau] - \widehat{h}_{\zeta,\tau}[\zeta]) < \sum_{j \notin \{\tau,\zeta\}} \lambda_j \widehat{h}_{\zeta,\tau}[j] Z[j] \\ \implies \lambda \widehat{h}_{\tau,\zeta}[\tau] \left((Z[\tau] - Z[\zeta]) + Z[\zeta] \frac{\widehat{h}_{\tau,\zeta}[\tau] - \widehat{h}_{\zeta,\tau}[\zeta]}{\widehat{h}_{\tau,\zeta}[\tau]} \right) < \sum_{j \notin \{\tau,\zeta\}} \lambda_j \widehat{h}_{\zeta,\tau}[j] Z[j]. \end{aligned} \tag{4.13}$$

We divide both sides by the quantity $\mathsf{CN}_{\tau,\zeta}$ defined as,

$$\mathsf{CN}_{\tau,\zeta} = \sqrt{\left(\sum_{j\notin\{\tau,\zeta\}} \lambda_j^2 (\widehat{h}_{\zeta,\tau}[j])^2\right)}.$$

This normalizes the RHS of (4.13) to have a standard normal distribution.

Thus, the necessary and sufficient condition for a misclassification error is for some $\zeta \neq \tau, \zeta \in [k]$,

$$\frac{\lambda \widehat{h}_{\tau,\zeta}[\tau]}{\mathsf{CN}_{\tau,\zeta}} \left((Z[\tau] - Z[\zeta]) + Z[\zeta] \frac{\widehat{h}_{\tau,\zeta}[\tau] - \widehat{h}_{\zeta,\tau}[\zeta]}{\widehat{h}_{\tau,\zeta}[\tau]} \right) < \frac{1}{\mathsf{CN}_{\tau,\zeta}} \sum_{j\notin\{\tau,\zeta\}} \lambda_j \widehat{h}_{\zeta,\tau}[j] Z[j]. \tag{4.14}$$

A sufficient condition for correct classification can then be obtained by ensuring that the smallest potential value of the LHS is still greater than the value of the RHS for all values of ζ. Thus, we obtain a sufficient condition for correct classification by appropriately minimizing or maximizing quantities over competing feature indices $\zeta \neq \tau, \zeta \in [k]$ (for notational convenience we simply denote this as $\min_\zeta$ or $\max_\zeta$).

$$\underbrace{\frac{\min_\zeta \lambda \widehat{h}_{\tau,\zeta}[\tau]}{\max_\zeta \mathsf{CN}_{\tau,\zeta}}}_{\text{SU/CN ratio}} \left(\underbrace{\min_\zeta (Z[\tau] - Z[\zeta])}_{\text{closest feature margin}} - \underbrace{\max_\zeta |Z[\zeta]|}_{\text{largest competing feature}} \underbrace{\cdot \max_\zeta \left| \frac{\widehat{h}_{\tau,\zeta}[\tau] - \widehat{h}_{\zeta,\tau}[\zeta]}{\widehat{h}_{\tau,\zeta}[\tau]} \right|}_{\text{survival variation}} \right)$$
$$> \underbrace{\max_\zeta \frac{1}{\mathsf{CN}_{\tau,\zeta}} \left(\sum_{j\notin\{\tau,\zeta\}} \lambda_j \widehat{h}_{\zeta,\tau}[j] Z[j] \right)}_{\text{normalized contamination}}. \tag{4.15}$$

We will show that under the conditions specified in Theorem 5, with sufficiently high probability, the relevant survival to contamination *SU/CN ratio* grows at a polynomial rate n^v for some $v > 0$, the *closest feature margin* shrinks at a less-than-polynomial rate $1/\sqrt{\ln nk}$, and the *survival variation* decays at a polynomial rate n^{-u} for some $u > 0$. Further, the magnitudes of the *largest competing feature* and the *normalized contamination* are no more than $2\sqrt{\ln(nk)}$. Here, we leverage the idea of typicality-style proofs in information theory [36] to avoid unnecessarily loose union bounds that end up being dominated by the atypical behavior of quantities. In our case, by pulling the feature margin out explicitly, we can just deal with its typical behavior. Similarly, the typical behavior of the largest competing feature and the true feature is all that matters.

4.3 Discussion

In this chapter, we compute sufficient conditions for good generalization of multiclass classification in a bi-level overparameterized linear model with Gaussian features. We observed

that multiclass classification can generalize even when the regression problem does not generalize (for $q + r > 1$). Further, the multiclass problem is "harder" than the binary problem because we have fewer positive training examples per class. The nature of the training data complicates our analysis in the multiclass setting since the true class labels are generated by comparing k features and thus we no longer have independence of the encoded class label y with any of these features. This becomes relevant when we compute bounds on the survival and contamination quantities since the Hanson-Wright inequality [124] is no longer applicable directly on the quantities of interest as was the case for the binary classification problem in Chapter 3. As a consequence of working around this non-independence we believe that our sufficient conditions for good generalization in the regime $q + r > 1$ are loose.

Even though in our work we focus on the regime where regression does not work, $q+r > 1$, we can extend the analysis to the regime where $q+r < 1$ by grinding through the expressions for survival and contamination in this regime. Even in this regime, for multiclass training data, survival is of the order $\frac{1}{k}$ while contamination scales similarly to the regime $q + r > 1$. Thus, while it is true that for binary classification or a fixed number of classes, the regime where regression works is a regime where classification also works, this need not be true if there are too many classes.

We conjecture that the following is a set of necessary and sufficient conditions for asymptotically good generalization (We elaborate on this in Section 4.4):

Conjecture 6. *(**Conjectured bi-level regions**): Under the bi-level ensemble model 9, when the true data generating process is a 1-sparse orthogonal means model (Assumption 2), as $n \to \infty$, the probability of misclassification event $\mathbb{P}(\mathcal{E}_{\text{err}})$ behaves as follows:*

$$\mathbb{P}(\mathcal{E}_{\text{err}}) \to \begin{cases} 0, & \text{if } t < \min\left(r, 1-r, p+1-2\cdot\max(1, q+r)\right) \\ 1, & \text{if } t > \min\left(r, 1-r, p+1-2\cdot\max(1, q+r)\right) \end{cases}. \tag{4.16}$$

The conjectured regions are visualized in (1d),(1e),(1f),(2d),(2e) and (2f) in Figure 4.1. Subfigures (1d) and (2d) illustrate that we believe multiclass classification with finitely many classes works if binary classification works. Further, comparing (1e) to (2e) when we increase q, the conjectured parameter region where multiclass classification works shrinks since we decrease the amount of favoring of true features. Interestingly, the nature of the looseness in our approach is such that our proof technique is able to recover a larger fraction of the conjectured region for larger q which intuitively is a result of less favoring leading to stronger concentration of certain random quantities. Tightening the potential looseness in our analysis and proving the converse result by computing sufficient conditions for poor generalization of multiclass classification are interesting avenues of future work.

Recent work from [146] provides an analysis of the generalization error of the minimum-norm interpolation of one-hot labels for multiclass classification with Gaussian features. While our work has many similarities with [146] in terms of model and problem setting, there are some key differences.

The first key difference is in how the training data is generated. In our work, we assume the true label of a point is generated based on which of the first k dimensions is the largest,

while [146] consider a Gaussian mixture model and a multinomial logistic model where the true labels have some randomness even conditioned on the first k dimensions. Like us, however, they also consider the case of orthogonal classes.

Second, we consider the asymptotic case where the number of classes, k, scales with the number of training points as $k = cn^t$ for some positive integer c and non-negative real t. The work in [146] considers only the fixed finite classes setting i.e. $t = 0$ in our model. The error analysis technique employed by us here in the form of a typicality-style argument featuring the feature margin (difference between the largest and second largest feature) is much tighter than the method employed in [146] and allows us to compute regimes where multiclass classification succeeds even when $t > 0$. A straight substitution into the analysis from [146] does not work since that analysis is too loose for this setting. Furthermore, in our expressions for survival and contamination (Lemmas 17 and 18) we compute an exact dependence on k.[7] The expressions from [146] don't compute this exact dependence because it is not required for their purposes. By using our novel analysis technique we are able to elucidate the challenges posed by fewer positive training examples per class in the multiclass setting and provide sufficient conditions for generalization when number of classes scales with the number of training points.

An equivalence between the solution obtained by minimum-ℓ_2-norm interpolation on the adjusted zero-mean one-hot encoded labels that we perform in our approach (4.6) and the solution obtained by other training methods has been established in [146]. In particular the minimum-norm interpolating solution is typically identical to the solution obtained via one-vs-all SVM and multi-class SVM (and thus gradient descent on cross-entropy loss due to its implicit bias [68, 131], under sufficient overparameterization. From [146], the sufficient conditions for the equivalence of solutions are,

$$\frac{\sum_{j=1}^{n} \lambda_j}{\lambda_1} > C_1 k^2 n \ln(kn),$$
$$\frac{(\sum_{j=1}^{n} \lambda_j)^2}{\sum_{j=1}^{n} \lambda_j^2} > C_2(\ln(kn) + n),$$

where C_1, C_2 are positive constants. Under our bi-level model (Definition 9) these conditions translate to:

$$q + r > 2t + 1,$$
$$2p - \max(2p - 2q - r, p) > 1,$$

[7] In particular, our analysis here brings out the fact that multiclass training data becomes less informative per training sample as the number of classes increases. This results in a $\frac{1}{k}$ scaling term in survival and a $\frac{1}{\sqrt{k}}$ scaling in contamination. It is this effect that makes it possible in some regimes for the contamination from other favored features to dominate — whereas in the case of binary classification, it is always the contamination from unfavored features that dominates.

which can be rearranged to give us the condition

$$0 < t < \frac{q + r - 1}{2}.$$

Figure 4.2 illustrates this regime, as well as how it relates to our results.

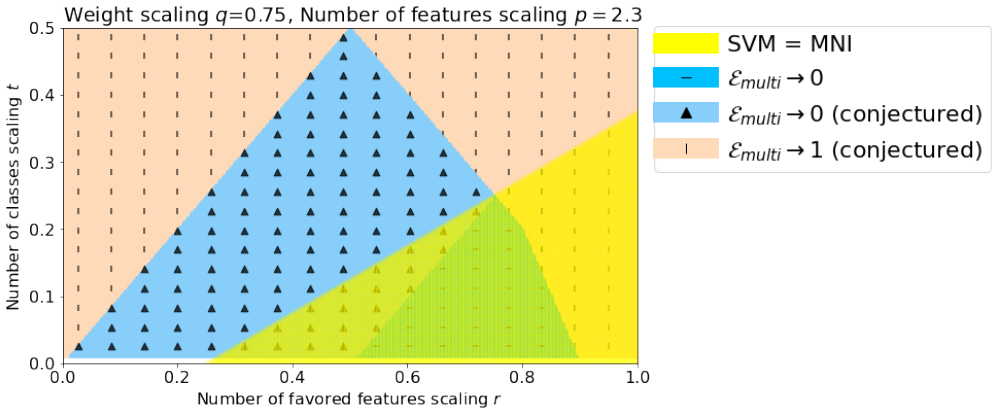


Figure 4.2: Visualization of regime where SVM solution is identical to MNI solution.

Notice the overlap. Thus our result is not limited only to the minimum-norm interpolator, but in fact holds for other training methods when the problem is sufficiently overparameterized. In this sense, the results in [146] and the present paper can be read together to tell a more full story of overparameterized multiclass classification. The behavior of the SVM solution in the conjectured region where $\varepsilon_{multi} \to 1$, but where it is not known whether SVM = MNI, is left for future work.

Further, although the present analysis focuses on solutions that exactly interpolate the training data, we can extend our results to account for additional ridge regularization by viewing ridge regularization as minimum-norm interpolation using augmented contamination-free features as in the Appendix of [105] and computing bounds leveraging tools from [141]. Our assumption of the strict bi-level weighting model is largely to simplify the calculations and by substituting terms appropriately in our lemmas from Section 4.8, it should be possible to compute results for other weighting models.

Finally, exploring the new phenomena that can be encountered as we go beyond the 1-sparse orthogonal means model is an exciting direction for future work. Here a possible first line of inquiry is whether multi-class classification can succeed when the number of classes k, exceeds $\sqrt{n}$ where n is the number of training points and the classes are determined by features that are *almost* orthogonal, i.e we are operating close to the 1-sparse orthogonal means model.

The next section discusses the potential looseness in our analysis and describes how we obtained Conjecture 6.

4.4 Conjectured looseness of bound

In (4.48) in the proof of Lemma 23, we upper bound $\mathbf{z}_j^\top \boldsymbol{\Delta} A_{inv} \boldsymbol{\Delta} y$ using the Cauchy-Schwarz inequality as

$$|\mathbf{z}_j^\top \boldsymbol{\Delta} A_{inv} \boldsymbol{\Delta} y| \leq \|\mathbf{z}_j\|_2 \|\boldsymbol{\Delta} A_{inv} \boldsymbol{\Delta} y\|_2 \tag{4.17}$$

$$\leq \|\boldsymbol{\Delta} A_{inv}\|_{op} \|\mathbf{z}_j\|_2 \|\boldsymbol{\Delta} y\|_2 \tag{4.18}$$

$$\leq \Delta_\mu \|\mathbf{z}_j\|_2 \|\boldsymbol{\Delta} y\|_2. \tag{4.19}$$

This results in a high-probability bound of the order $\Delta_\mu n/\sqrt{k}$. Essentially this bound fears that $\boldsymbol{\Delta} A_{inv}$ can, in worst case, align $\mathbf{z}_j$ and $\boldsymbol{\Delta} y$ to be in the same direction. However, since there is only a weak dependence between $\boldsymbol{\Delta} A_{inv}$ and $\mathbf{z}_j$ and $\boldsymbol{\Delta} y$ this bound is likely overly cautious. We conjecture that this bound is loose by a factor $\sqrt{n}$. Why do we conjecture this? If we ignored the dependency of $\boldsymbol{\Delta} A_{inv}$ on $\mathbf{z}_j$ and $\boldsymbol{\Delta} y$ and blindly applied the Hanson-Wright inequality (with the $\mathbf{M}$ matrix introduced as in Section 4.10 to leverage the fact that $\boldsymbol{\Delta} y$ is mostly zeros) then we would obtain a high-probability upper bound of the form $\Delta_\mu \sqrt{n/k}$ (ignoring the logarithmic factors).

Assuming this tighter conjectured bound holds and similarly assuming an analogously tighter bound for $\left|\mathbf{z}_\tau^\top \boldsymbol{\Delta} A_{inv} \boldsymbol{\Delta} y\right|$ in Section 4.10 and following through with the rest of our analysis, we obtain the conjectured sufficient conditions for good generalization as in Equation (4.16) from Conjecture 6 for the regime $q + r > 1$.

It turns out that whenever the survival/contamination ratio grows at a polynomial rate n^v for $v > 0$ then the survival variation term also shrinks at a polynomial rate n^{-u} for $u > 0$. Thus ensuring the survival/contamination ratio is large enough (i.e. the number of classes is not too large relative to the level of favoring of potentially true features) is key to obtaining good generalization.

Although we focus on the regime $q + r > 1$ in our work, our proof technique is also applicable to the regime $q + r < 1$, i.e where regression works and by grinding through the math for this setting we should be able to get sufficient conditions for good generalization here as well. The survival in the multi-class setting in the regime $q+r < 1$ will scale roughly as $1/k$ due to the fewer positive training examples per class instead of behaving like the constant $\sqrt{2/\pi}$ as was the case for binary classification (Lemma 9, Chapter 3). Moreover, Lemma 11 from Chapter 3 shows that for the binary classification setting the contamination scales as $n^{-\min(p-1,1-r)/2}$ when $q+r \leq 1$. In the multiclass setting the contamination will be lower by a factor of $\sqrt{k}$ and substituting this in our error analysis we obtain Conjecture 6 for the regime $q + r < 1$.

Finally, we believe that we can adapt our analysis from the Proof of Theorem 5 in Section 4.8 to write a set of sufficient conditions for poor generalization. The primary condition for this would be for the relevant survival/contamination ratio to go to zero. We conjecture that computing conditions on p, q, r, t under which this occurs results in the converse result in the form of sufficient conditions for poor generalization present in Conjecture 6. Intuitively, if the survival/contamination ratio goes to zero, then the contamination can with significant

probability flip the sign of a comparison involving the score that should be winning — this parallels the way that the converse is proved in Chapter 3 for binary classification.

The next section elaborates on the effect of fewer number of positive training examples per class in the multiclass setting and investigates an alternative setting where the total number of positive training examples per class is kept constant while we increase the number of classes.

4.5 Scaling parameters with the number of positive training examples per class

From our results in Figure 4.1 we observed that as the number of classes k increases (i.e. larger values of t), the region where multiclass classification generalizes well shrinks. A justification for this is when the number of classes k increases while the number of training points n stays constant, we have fewer positive training examples from each class, and this makes the task harder.

To see if the reduced number of positive training examples is indeed the dominant effect, we can explore what happens if we increase the number of total training points to compensate for this effect? Instead of scaling all parameters with the total number of training points, what happens if we scale them with the number of positive training examples per class?

Let $N = n^b$ be the new number of training points for some $b > 1$, while rest of the parameters in the bi-level model scale as before. We have,

$$\begin{aligned} N &= n^b \\ d &= n^p = N^{p/b} \\ s &= n^r = N^{r/b} \\ a &= n^{-q} = N^{-q/b} \\ k &= c_k n^{-t} = c_k N^{-t/b}. \end{aligned}$$

We can interpret this as our standard setup, albeit parameterized by N, rather than n. To keep the model well-defined we require the following:

- $b < p$, to ensure we are still overparameterized;
- $r < b$, to ensure the number of favored features does not exceed the total number of training points;
- $q < p - r$ to ensure we are actually favoring the first s features.

For this setup, Theorem 5 states that the probability of misclassification tends to zero if

$$\begin{aligned} \frac{t}{b} &< \min\left(\frac{r}{b}, 1-\frac{r}{b}, \frac{p}{b}+1-2\left(\frac{q}{r}+\frac{r}{b}\right), \frac{p}{b}-2, \frac{2q}{b}+\frac{r}{b}-2\right) \\ \frac{q}{b}+\frac{r}{b} &> 1. \end{aligned}$$

Rearranging, we obtain the condition

$$\begin{aligned} t &< \min(r, b-r, p+b-2(q+r), p-2b, 2q+r-2b) \\ q+r &> b. \end{aligned}$$

To hold the number of training samples per class fixed we can set $b = t+1$, so the ratio N/k becomes constant. Doing so, we obtain the following sufficient conditions for good generalization:

$$\begin{aligned} t &< \min\left(r, \frac{p-2}{3}, \frac{2q+r-2}{3}\right) \\ 0 &< 1-r \\ 0 &< p+1-2(q+r) \\ t &< q+r-1. \end{aligned}$$

Additionally for the model to be well defined we require $t < p-1$. (The other conditions $r < t+1$ and $q < p-r$ for model to be well defined are automatically satisfied if the above conditions for good generalization are satisfied).

If we assume Conjecture 6 then a set of sufficient conditions for good generalization is:

$$\begin{aligned} 0 &\leq p+1-2(q+r) \\ r &< 1 \\ t &< r \\ t &< p-1. \end{aligned}$$

The first two conditions must be satisfied for binary classification problem to generalize well and thus for multi-class classification to succeed in this setting we need to ensure binary classification succeeds. The condition $t < r$ arises because if we don't favor the features used in the comparison while assigning class labels then we have no hope of succeeding in overparameterized settings. The condition $t < p-1$ ensures that the problem is overparameterized. If any of these conditions is not met then the probability of classification error will tend to 1.

Figure 4.3 visualizes the conjectured regimes for this alternative setup where the number of positive training examples per class is held fixed as we vary the number of classes for fixed values of p and q. In the white region, our model is not well defined. Note that in subfigure (a), the limiting factor to the model being well defined is the inequality $r < 1+t$

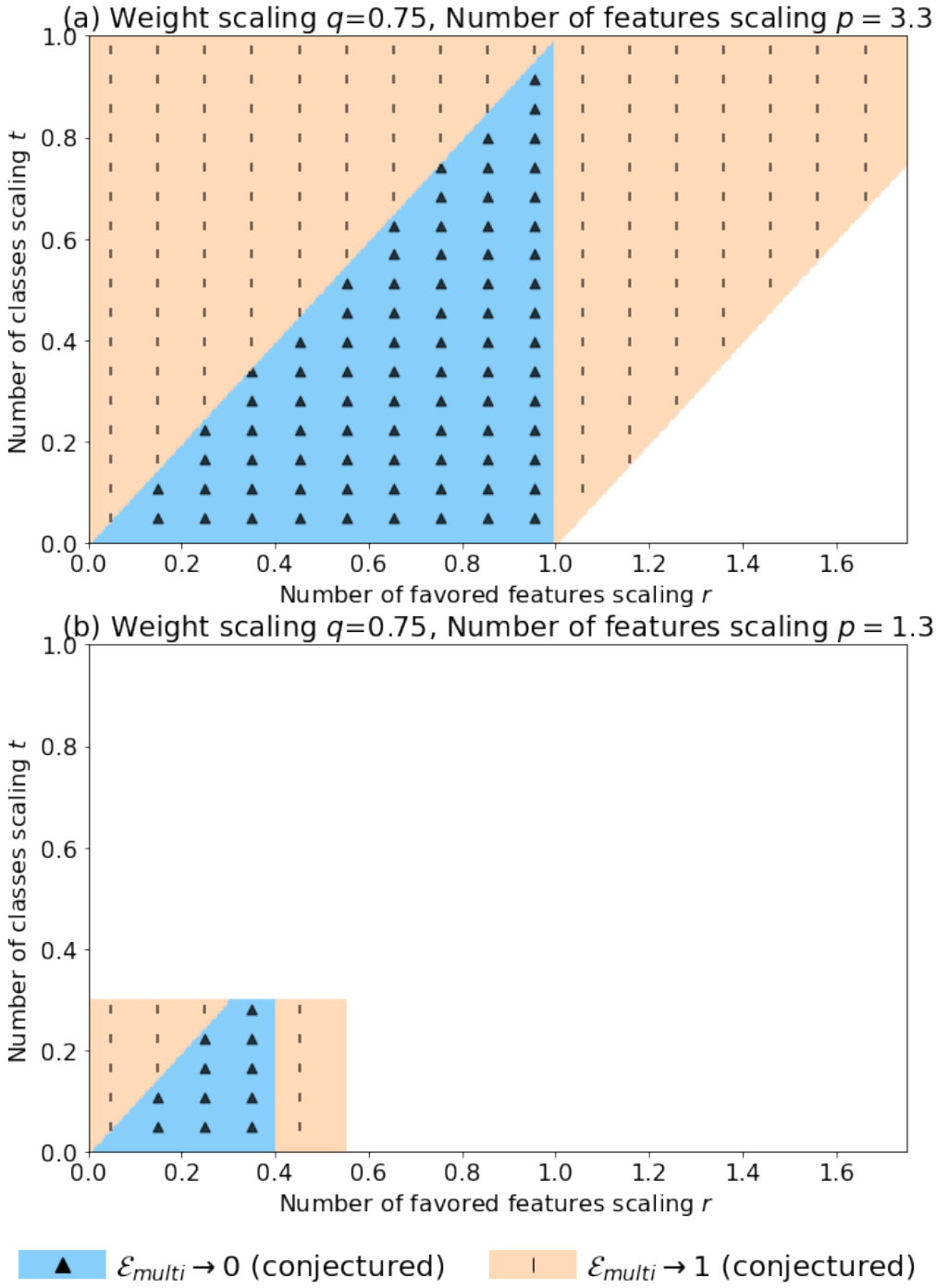


Figure 4.3. Visualization of the conjectured bi-level classification regimes when we scale everything with the number of positive training examples per class, instead of with the total number of training points.

(we must have more training examples than favored features) while in subfigure (b), the limiting factor for the model being well defined in the right-hand boundary is the inequality $r < p - q$ (we must put a larger weight on the features we favor as compared to those that we do not favor). In subfigure(b) we see that the top boundary for the model being well defined is the inequality $t < p-1$ which is necessary for the problem to be overparameterized and support the existence of interpolating solutions. Further, the right-hand bound for good generalization in subfigure (a) corresponds to the inequality $r < 1$ while in subfigure (b) it corresponds to $p + 1 > 2(q + r)$. The left-hand boundary for good generalization in both figures is the inequality $t < r$, which reflects the fact that for MNI-based classification to succeed, all the features defining the classes must be favored.

It is interesting to note that when we add more training points so as to increase the number of positive examples, we are effectively decreasing the level of overparameterization in the problem. We know from [109] that adding training data in a way that reduces overparameterization can sometimes make performance worse instead of better. However, in the deeply overparameterized setting of the bi-level models explored here, this effect is counteracted by the survival benefits of having more positive examples — in effect, reducing the overall level of overparameterization reduces the shrinkage induced by the regularizing effect of overparameterization. This reduction in shrinkage compensates for the $\frac{1}{k}$ hit to survival induced by the larger number of classes.

The next section complements the theoretical and asymptotic proofs in this chapter with an empirical evaluation of relevant quantities using simulated data for finite values of , the number of training points. We chose a regime where we are conjecturing results so that it is possible to see the very close match between our conjectured predictions for how these quantities should scale and how they actually do in experiments.

4.6 Experimental results

Theorem 5 is proved rigorously and so we know that the asymptotic result is true. Underlying the result is the analysis of survival (how strongly is the true feature underlying this class represented in the learned score) and contamination (what is the standard deviation of the contamination in predictions that comes from learning nonzero coefficients to features that have nothing to do with this class). Multiclass classification asymptotically succeeds when the survival dominates the contamination.

In Figure 4.4, we plot experimental results using the bi-level ensemble model (Definition 9) for a setting where regression does not work but multiclass classification is conjectured to work. We plot quantities from Equation (4.15) in our error analysis. From subfigures (a),(b) and (c) we observe that while both survival and contamination are decreasing as we increase n, the survival/contamination ratio increases. The survival/contamination ratio growing with n is important for correct classification. The trend is very clear from the experimental results and indicates that continuing to grow n (together with the number of classes k, the number of features d, the number of favored features s, and the level of favoring as per our

bi-level idealized model) would result in ever improving performance. Furthermore, we see that the empirical slope of these quantities on a log-log plot (and thus the power-law scaling of these quantities with respect to n) agree with the theoretical slopes calculated based on our conjecture.

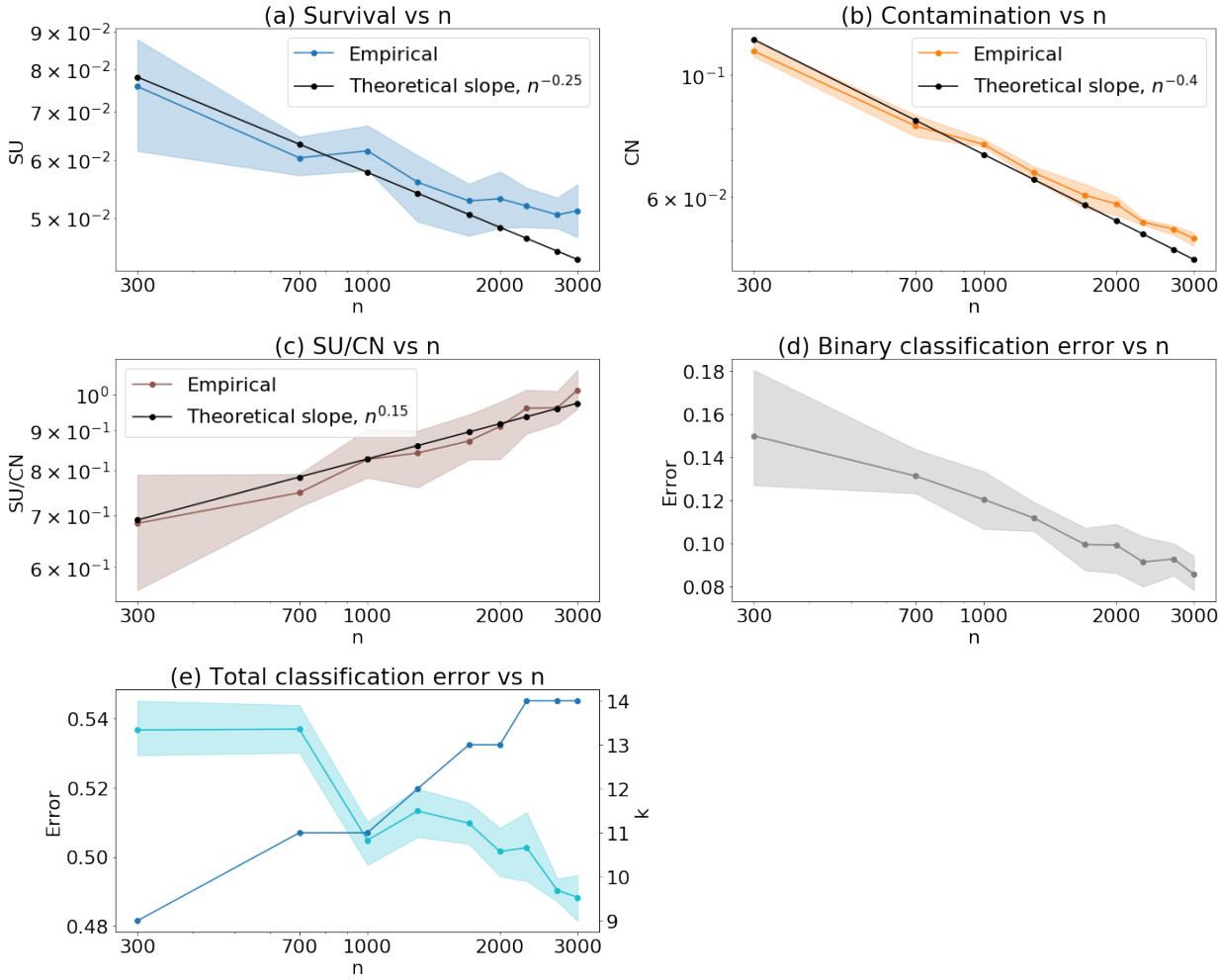


Figure 4.4. Experimental results using the bi-level ensemble model with $p = 1.5, q = 0.55, r = 0.5, t = 0.2$. Here, the number of training samples n varies from 300 to 3000 and the number of classes is computed as $k = \lfloor 3n^t \rfloor$ and varies from 9 to 14. We calculated the classification errors over a batch size of 10000, and ran 10 trials. The plots show the mean plotted with error bars corresponding to the $10^{th}, 90^{th}$ percentile values. We also plot the theoretical slopes for survival, contamination and the survival/contamination ratio based on our conjecture and notice that it closely matches the empirical slope of the quantities when plotted on a log-log scale. Notice that jaggedness in the plots is often due to integer effects as k grows or does not grow with n.

Subfigure (d) plots the binary classification error when only trying to distinguish between the true class and one other particular class. (In this experiment, the true class was deter-

mined by feature 1. We calculate the binary error as the probability of misclassifiying a point from class 1 as belong to class 2 when we only compare the scores for class 1 and class 2.) We see that this error clearly decreases as we increase n. One way of thinking about successful multiclass classification is that the true class must win such pairwise competitions against all competing classes. Finally, subfigure (e) plots the total multiclass misclassification error overlaid with the number of classes. Here too we see a downward trend in classification error as n increases, even though we would have to go to significantly larger n than our compute could handle to see this error probability drop to very low values. Notice the integer effects arising from the number of classes k sometimes not growing with n that result in small upward spikes in the classification error.

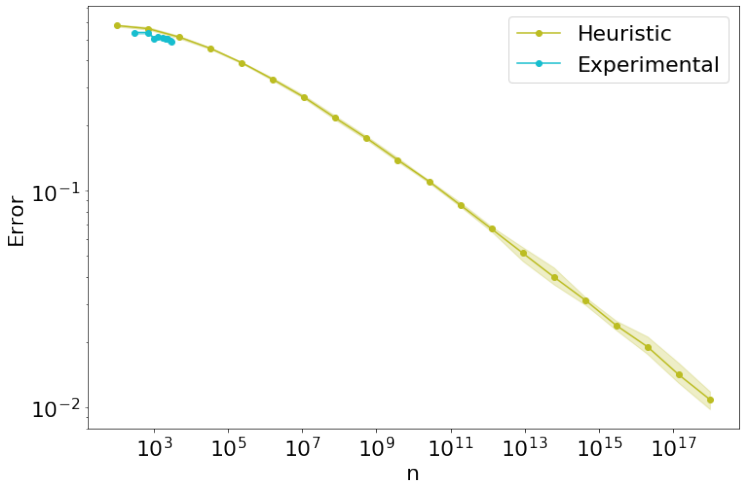
To estimate how large n must be to see the probability of misclassification get close to zero for the settings as in the experiment from Figure 4.4, we perform a heuristic calculation where the scores used to predict the class on a test point (actually generated by min-norm interpolation process) are instead determined as sum of the scaled version of the feature (down-scaled by the survival) and and independent zero mean Gaussian with standard deviation equal to contamination. Using the curves from subfigures(a) and (b) of Figure 4.4 we can extrapolate the survival and contamination quantities as:

$$\mathsf{SU} = 0.32 \cdot n^{-0.25}, \quad \mathsf{CN} = \frac{1.13}{\sqrt{2}} \cdot n^{-0.4}, \tag{4.20}$$

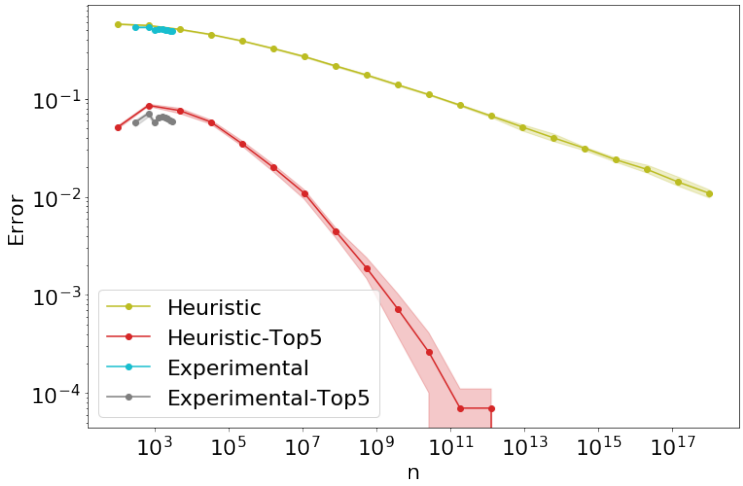
where we down-scaled contamination by $\sqrt{2}$ since Figure 4.4 plots corresponds to contamination from two features (the competing feature and the true feature).

Figure 4.5 plots the classification error as well as the "Top-5" error (the probability that the predicted class corresponds to one of the five largest features) obtained by our heuristic calculation as well as our experiment. We see that for n where we were able to computationally run the experiment the error values are close to the heuristic calculation. The power of the heuristic calculation (and of our theory to enable such heuristic calculations) is that we can predict what happens for very large n. For instance, we need $n \approx 10^{18}$ for the classification error to drop below 0.01. If we look at the Top-5 error however then $n \approx 10^7$ is sufficient for error to go below 0.01. This corresponds to 10^{10} parameters (since $d = n^{1.5}$) and while this is clearly too large for us to run on our local machines, dedicated GPU clusters given sufficient time are capable of running this. To conclude, even though for the parameter settings we considered above, the classification error doesn't drop close to zero until n becomes very large, we see the power of our theoretical result in enabling heuristic calculations to predict the evolution of classification error with the number of training samples.

The next section plots the difference between the singular values (and hence estimated eigenstructure) of the empirical feature matrix and the eigenstructure of the underlying features themselves. While this behavior is well known in the literature, the plots illustrate the underlying challenge provided by the regime in which regression does not generalize — namely that the empirical eigenstructure does not reveal the true nature of the underlying features.



(a) Classification error vs number of training points



(b) Top-5 error and classification error vs number of training points

Figure 4.5. Heuristic calculation of multiclass classification error for $p = 1.5, q = 0.55, r = 0.5, t = 0.2$. Here, the number of training samples n varies from 10^2 to 10^{18} and the number of classes is computed as $k = \lfloor 3n^t \rfloor$ and varies from 7 to 11943. We compute the heuristic classification error and Top-5 error over a batch size of 10000, and ran 10 trials. The plots show the mean plotted with error bars corresponding to the $10^{th}, 90^{th}$ percentile values.The heuristic calculations are a close match for the experimental values from Figure 4.4. For $n > 10^{11}$, the heuristic calculation for Top-5 error is 0 since our batch size is not large enough to detect errors smaller than 10^{-4}.

4.7 Comment on empirical eigenstructures of feature matrices

It is well known (for instance Remark 1 from Chapter 3 that cites [147]) that for a spiked covariance model when the ratio of the top to the bottom eigenvalues grows as $\Omega(d/n)$, the top s eigenvalues can be estimated reliably from samples, even when the number of training samples n is less than the number of features d. The ratio of the top to the bottom eigenvalue in our bi-level model scales as

$$\frac{\frac{ad}{s}}{\frac{(1-a)d}{d-s}} = n^{p-q-r}, \tag{4.21}$$

and when $q + r < 1$, this ratio is larger than $d/n = n^{p-1}$. Figure 4.6 shows empirical results of estimating the eigenvalues via the singular value decomposition of the training feature matrices. The visual distinction is quite striking. In the regime $q + r > 1$, the SVD of the training features matrix (and thus the empirical covariance matrix's eigenvalues) does not reveal that there are actually s favored features in the data. By contrast, in the regime $q + r < 1$, the SVD clearly shows an eigenvalue gap that reveals exactly what s is.

Theorem 5 shows that when $q + r > 1$ and there is not enough structure in the feature matrix to reliably estimate the top eigenvalues of the feature covariance matrix features, multiclass classification can still succeed for interpolating solutions as long as the number of classes does not increase too fast. It is because we are in the regime $q + r > 1$ that new techniques for analysis had to be developed in this chapter, and the gap between the regime where we can prove the results and our conjectured results points interestingly to where there is a need for even better technique.

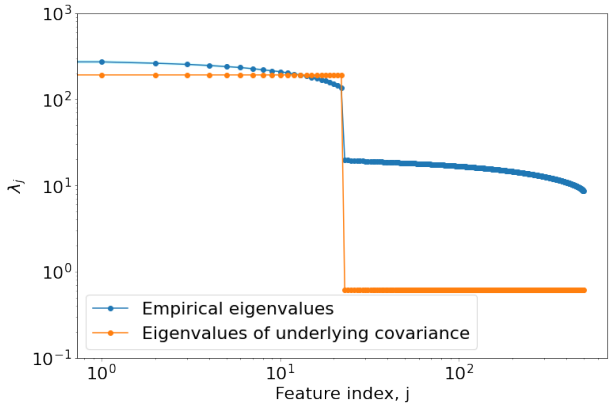
The rest of the chapter is organized as follows. Section 4.8 provides an overall proof for Theorem 5 by introducing some intermediate lemmas and assuming they hold. Section 4.9 introduces some key tools that we need and Section 4.10 leverages those tools to build towards a proof of these intermediate lemmas by introducing some helper results that are needed to deal with the key challenge posed by multiclass training data. Section 4.11 actually proves the intermediate lemmas used in Section 4.8 and completes the proof.

Throughout, we will assume that n is large enough for asymptotic behavior to kick in. We also will introduce various universal positive constants, indexed as c_i. These constants are all independent of n, and constants with the same index are to be treated as equal.

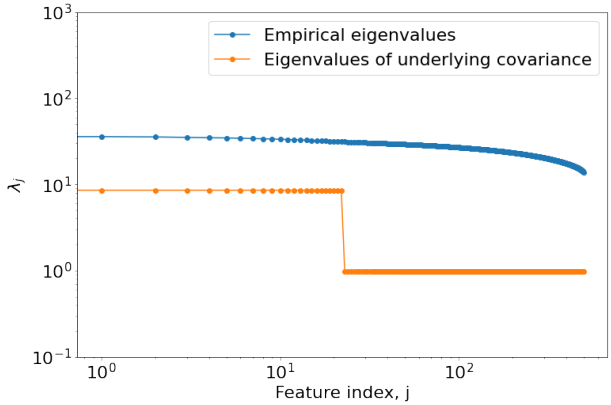
4.8 Appendix: Proof of Theorem 5

We restate Theorem 5, our main result, here for convenience:

Theorem 5. *(**Asymptotic classification region in the bi-level model**): Under the bi-level ensemble model 9, when the true data generating process is from a 1-sparse orthogonal*



(a) $q = 0.15, q + r = 0.65$.



(b) $q = 0.65, q + r = 1.15$.

Figure 4.6. Estimating the eigenvalues of the covariance matrix of features empirically. Here $n = 400$ and the feature covariance structure follows the bi-level model with parameters $p = 1.5, r = 0.5$. Thus $d = 8001$ and $s = 21$ for both (a) and (b). The difference between (a) and (b) is in the level of favoring of favored features: with $q = 0.15$, (a) favors them more than (b) does with $q = 0.65$. In the regime where regression works, $q + r \leq 1$, we are able to accurately estimate the top s eigenvalues. In the regime where regression fails, $q + r > 1$, we are unable to estimate the top s eigenvalues accurately. The blue curve plots the estimated eigenvalues and the shaded region corresponds to the 10-90 percentile of the estimated values over 20 trials. Note, that there is only very small deviation across trials.

means model (Assumption 2), the probability of misclassification $\mathbb{P}(\mathcal{E}_{\text{err}}) \to 0$ *as* $n \to \infty$ *if the following conditions hold:*

$$t < \min(r, 1 - r, p + 1 - 2(q + r), p - 2, 2q + r - 2)$$
$$q + r > 1.$$

Before we proceed with the proof we remind the reader of a few important definitions.

Recall from (4.7) that our learned feature coefficients are

$$\widehat{\boldsymbol{\alpha}}_m = (\boldsymbol{\Phi}_{\mathsf{train}})^\top \left(\boldsymbol{\Phi}_{\mathsf{train}}(\boldsymbol{\Phi}_{\mathsf{train}})^\top\right)^{-1} \mathbf{y}_m.$$

Let

$$\mathbf{A} = \boldsymbol{\Phi}_{\mathsf{train}}(\boldsymbol{\Phi}_{\mathsf{train}})^\top.$$

We can express $\mathbf{A}$ as,

$$\mathbf{A} = \sum_{j=1}^{d} \lambda_j \mathbf{z}_j \mathbf{z}_j^\top,$$

where $\mathbf{z}_j$ are i.i.d with $\mathbf{z}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$. The learned coefficients can then be written as

$$\widehat{\alpha}_m[j] = \sqrt{\lambda_j} \mathbf{z}_j^\top \mathbf{A}^{-1} \mathbf{y}_m.$$

Let $\mu_1(\mathbf{A})$ denote the largest eigenvalue and $\mu_n(\mathbf{A})$ denote the smallest eigenvalue of $\mathbf{A}$ respectively, with $\mu_i(\mathbf{A})$ being the i-th largest eigenvalue of $\mathbf{A}$.

Next, we state a useful lemma adapted from [10] that bounds the eigenvalues of $\mathbf{A}^{-1}$ and is a stronger version of Lemma 1 from Chapter 3. Subsequent lemmas will utilize these eigenvalue bounds.

Lemma 14. *(**Eigenvalue bounds on $\mathbf{A}^{-1}$ adapted from [10]**): If $\boldsymbol{\Lambda}$ is such that $\lozenge \ll \sum_j \lambda_j$, then with probability at least $(1 - 2e^{-n})$,*

$$\bar{\mu} - \Delta_\mu \leq \mu_n(\mathbf{A}^{-1}) \leq \mu_1(\mathbf{A}^{-1}) \leq \bar{\mu} + \Delta_\mu,$$

where,

$$\bar{\mu} = \frac{1}{\sum_j \lambda_j} \tag{4.22}$$

$$\lozenge = \frac{32}{9}\left(\lambda_1(1 + \ln 9)n + \sqrt{(1 + \ln 9)n \sum_j \lambda_j^2}\right) \tag{4.23}$$

$$\Delta_\mu = \bar{\mu}\left(\frac{\lozenge}{\sum_j \lambda j} + \Theta\left(\frac{\lozenge}{\sum_j \lambda j}\right)^2\right). \tag{4.24}$$

Further this implies that with probability at least $(1 - 2e^{-n})$,

$$\left|\mu_i(\mathbf{A}^{-1} - \bar{\mu}\mathbf{I}_n)\right| \leq \Delta_\mu$$

for all $i \in [n]$.

The subsequent lemmas bound the feature margin, survival, contamination and survival variation terms, utilizing tools from [10] and building on results from Chapter 3.

Lemma 15. *(**Lower bound on the closest feature margin as** $k \to \infty$): For any constant $\varepsilon > 0$, there exists a constant θ such that, for sufficiently large k with probability at least $(1-\varepsilon)$,*

$$\min_{\zeta:1\le\zeta\neq\tau\le k}(Z[\tau]-Z[\zeta]) \ge \frac{\theta}{\sqrt{2\ln(k)}}.$$

Here, τ is fixed and corresponds to the index of the true class — i.e. τ corresponds to the index of the maximum feature among the first k features.

Lemma 16. *(**Lower bound on the closest feature margin when** k **is constant**): If $k = c_k$ for some fixed constant c_k, for any constant $\varepsilon > 0$, there exists a constant $\varepsilon' > 0$ such that*

$$\mathbb{P}\left(\min_{\zeta,\gamma:1\le\zeta\neq\gamma\le c_k}|Z[\zeta]-Z[\gamma]| \ge \varepsilon'\right) \ge 1-\varepsilon.$$

Thus, with probability at least $(1-\varepsilon)$,

$$\min_{\zeta:1\le\zeta\neq\tau\le k}(Z[\tau]-Z[\zeta]) \ge \varepsilon'.$$

Here, τ is fixed and corresponds to the index of the true class — i.e. τ corresponds to the index of the maximum feature among the first k features.

Lemma 17. *(**Lower bound on relative survival of true feature**): For any fixed $\zeta \in [k]$, $\zeta \neq \tau$, with $\lambda_\tau = \lambda_\zeta = \lambda$ we have with probability at least $(1-5/(nk))$,*

$$\lambda\widehat{h}_{\tau,\zeta}[\tau] \ge \lambda\left(c_{10}\bar{\mu}\frac{n}{k}\sqrt{\ln(k)} - c_9(\bar{\mu}\sqrt{n}\sqrt{\ln(nk)} + \Delta_\mu\cdot n/\sqrt{k})\right),$$

for universal positive constants c_9 and c_{10}.

By substituting the asymptotic behavior of parameters from our bi-level ensemble model we get the following corollary:

Corollary 2. *Under the bi-level ensemble model 9, for any fixed $\zeta \in [k]$, $\zeta \neq \tau$, $\lambda_\tau = \lambda_\zeta = \lambda$ if $t < 1/2$, $t < 2(q+r-1)$ and $1 < q+r < (p+1)/2$, with probability at least $(1-5/(nk))$,*

$$\lambda\widehat{h}_{\tau,\zeta}[\tau] \ge c_{12}n^{1-q-r-t}\sqrt{\ln(k)},$$

for universal positive constant c_{12}.

Lemma 18. *(**Upper bound on contamination**): For any fixed $\zeta \in [k]$, $\zeta \neq \tau$, with probability at least $(1 - 7/(nk))$,*

$$\mathsf{CN}_{\tau,\zeta} \leq c_7 \left(\bar{\mu} \sqrt{\frac{n}{k}} \cdot \sqrt{\ln(ndk)} + \Delta_\mu \cdot \frac{n}{\sqrt{k}} \right) \cdot \sqrt{\sum \lambda_j^2},$$

for universal positive constant c_7.

As before, for our bi-level ensemble model we have the corollary:

Corollary 3. *Under the bi-level model 9, in the regime $1 < q+r < (p+1)/2$, with probability at least $(1 - 7/(nk))$,*

$$\mathsf{CN}_{\tau,\zeta} \leq c_{13} n^{(1-t-p)/2+\max(0,3/2-q-r)+\max(0,p/2-q-r/2)} \sqrt{\ln(ndk)},$$

for universal positive constant c_{13}.

Lemma 19. *(**Upper bound on survival variance**): For any fixed competing feature $\zeta \in [k]$, $\zeta \neq \tau$ with $\lambda_\tau = \lambda_\zeta$, we have with probability at least $(1 - 15/(nk))$,*

$$\frac{\widehat{h}_{\tau,\zeta}[\tau] - \widehat{h}_{\zeta,\tau}[\zeta]}{\widehat{h}_{\tau,\zeta}[\tau]} \leq \frac{2c_9(\bar{\mu}\sqrt{n}\sqrt{\ln(nk)} + \Delta_\mu \cdot n/\sqrt{k})}{c_{10}\bar{\mu}\frac{n}{k}\sqrt{\ln(k)} - c_9(\bar{\mu}\sqrt{n}\sqrt{\ln(nk)} + \Delta_\mu \cdot n/\sqrt{k})}, \tag{4.25}$$

for universal positive constants c_9 and c_{10}.

As before, we can also obtain the asymptotic bound:

Corollary 4. *Under the bi-level ensemble model 9, for any fixed $\zeta \in [k]$, $\zeta \neq \tau$, if $t < 1/2$, $t < 2(q + r - 1)$, and $1 < q + r < (p + 1)/2$, with probability at least $(1 - 15/(nk))$,*

$$\frac{\widehat{h}_{\tau,\zeta}[\tau] - \widehat{h}_{\zeta,\tau}[\zeta]}{\widehat{h}_{\tau,\zeta}[\tau]} < n^{-u},$$

for large enough n for some fixed $u > 0$.

Substitute Corollaries 2, 3, and 4 into (4.15), applying them on all $1 \leq \zeta \neq \tau \leq k$. They hold with probability at least $1-5/(nk)$, $1-7/(nk)$, and $1-15/(nk)$ respectively for a given test point and choice of ζ. So by the union bound across the three bounds and all $k - 1$ choices of ζ, with probability at most $27/n$, one of these corollaries will not hold for our test point for some ζ. Let this failure event be denoted E_1.

In the case when E_1 does not occur, misclassification occurs only if

$$\frac{c_{12}\sqrt{\ln(k)}}{c_7\sqrt{\ln(ndk)}} n^v \left(\min_\zeta (Z[\tau] - Z[\zeta]) - \max_\zeta |Z[\zeta]| \cdot n^{-u} \right) < \max_\zeta G^{(\zeta)},$$

where we define the exponent

$$\begin{aligned} v &= 1 - q - r - t - (1 - t - p)/2 - \max\left(0, \frac{3}{2} - q - r\right) - \max\left(0, \frac{p}{2} - q - \frac{q}{2}\right) \\ &= \frac{p+1}{2} - q - r - \frac{t}{2} - \max\left(0, \frac{3}{2} - q - r, \frac{p}{2} - q - \frac{r}{2}, \frac{3}{2} - 2q - \frac{3r}{2}\right), \end{aligned}$$

and

$$G^{(\zeta)} = \frac{1}{\mathsf{CN}_{\tau,\zeta}} \left(\sum_{j \notin \{\tau,\zeta\}} \lambda_j \widehat{h}_{\zeta,\tau}[j] Z[j] \right).$$

For each class ζ, observe that we have $G^{(\zeta)} \sim \mathcal{N}(0,1)$.[8] Thus, by the Gaussian tail bound, for each ζ with probability at least $(1 - 1/(nk))$,

$$G^{(\zeta)} < \sqrt{2\ln(nk)}. \tag{4.26}$$

So by the union bound over all k classes ζ, with probability at least $(1 - 1/n)$,

$$\max_{\zeta} G^{(\zeta)} < \sqrt{2\ln(nk)}.$$

Let the failure event where this is not the case be E_2.

An identical argument shows that with probability at least $(1 - 2/n)$, $\max_{\zeta} |Z[\zeta]| \leq \sqrt{2\ln(nk)}$. Let E_3 be the failure event where this is not the case.

From Lemma 15, we know with probability $1 - \varepsilon$ that, if $t > 0$, then for sufficiently large n (and so sufficiently large k)

$$\min_{\zeta} \left(Z[\tau] - Z[\zeta]\right) > \frac{\theta}{\sqrt{2\ln(k)}}.$$

If $t = 0$ and $k = c_k$, then Lemma 16 states that, with probability $1 - \varepsilon$,

$$\mathbb{P}\left(\min_{1 \leq \zeta \neq \gamma \leq c_k} |Z[\zeta] - Z[\gamma]| \geq \varepsilon' \right) \geq 1 - \varepsilon,$$

for some constant ε'. Let the ε-probability event of the appropriate margin bound (depending on whether $t = 0$ or $t > 0$) being violated be the error event E_4.

Assuming E_1, E_2, E_3, and E_4 all do not take place, misclassification can only occur if

$$\frac{c_{12}\sqrt{\ln(k)}}{c_7\sqrt{\ln(ndk)}} n^v \left(\min\left(1 - \varepsilon, \frac{\theta}{\sqrt{2\ln(k)}}\right) - \sqrt{2\ln(nk)} n^{-u} \right) < \sqrt{2\ln(nk)}.$$

[8] To be precise, here we can think of fixing the training data and looking purely at the randomness arising from the features in the test point. The resulting $G^{(\zeta)}$ is a standard normal. Since we are using the union bound in our proof finally, this is sufficient for our purposes.

Clearly, if $v > 0$, then (for sufficiently large n) misclassification becomes asymptotically impossible (except via the specified error events), since the LHS of the above grows asymptotically faster than the RHS.

The union bound shows that the probability of any of E_1, E_2, E_3, E_4 occurring tends to ε as $n \to \infty$ (since the probability of the first three tend to zero). So in the regime where

$$\begin{aligned} t &< \frac{1}{2} \\ t &< 2(q+r-1) \\ q+r &> 1 \\ \frac{p+1}{2} &> q+r+\frac{t}{2}+\max\left(0,\frac{3}{2}-q-r\right)+\max\left(0,\frac{p}{2}-q-\frac{r}{2}\right), \end{aligned} \tag{4.27}$$

the probability of misclassification tends to ε for sufficiently large n, for any $\varepsilon > 0$.

Consolidation of the above bounds produces the conditions [9]

$$\begin{aligned} t &< \min\left(1-r, p+1-2(q+r), p-2, 2q+r-2\right) \\ q+r &> 1. \end{aligned}$$

Finally, note that the condition $t < r$ comes from the definition of the bi-level model (9). This condition simply states that for good generalization we must favor all the features used to determine classes. Since the analysis above holds for any ε, we see that within this regime the probability of misclassification must approach zero in the limit. This completes the proof. Note that while we show that probability of misclassification goes to zero, we do not show it to do so at any particular rate, because the result from Lemma 15 does not specify the rate of convergence.

4.9 Appendix: Useful results from elsewhere that we need

This section collects results that are used in our proof, but which come from elsewhere or are lightly adapted to our purposes.

[9] We can simplify (4.27) as follows:

$$\begin{aligned} \frac{p+1}{2} > q+r+\frac{t}{2} &\implies t < p+1-2(q+r) \\ \frac{p+1}{2} > q+r+\frac{t}{2}+\frac{3}{2}-q-r &\implies t < p-2 \\ \frac{p+1}{2} > q+r+\frac{t}{2}+\frac{p}{2}-q-\frac{r}{2} &\implies t < 1-r \\ \frac{p+1}{2} > q+r+\frac{t}{2}+\frac{3}{2}-q-r+\frac{p}{2}-q-\frac{r}{2} &\implies t < 2q+r-2. \end{aligned}$$

Then we note that $t < \min(r, 1-r) \implies t < 1/2$.

The first result is the Hanson-Wright inequality that we saw previously as Lemma 2 in Chapter 3. We restate a more general form here for convenience where the sub-Gaussian norm of the random variables of interest is arbitrary.

Hanson-Wright inequality [124]: Let $\mathbf{z}$ be a random vector composed of i.i.d. random variables that are zero mean and with sub-Gaussian norm at most K. The sub-Gaussian norm $\|\xi\|_{\psi_2}$ of a random variable ξ is defined as in [124],

$$\|\xi\|_{\psi_2} = \inf_{K>0} K \tag{4.28}$$

$$\text{s.t. } \mathbb{E}\exp\left(\xi^2/K^2\right) \leq 2. \tag{4.29}$$

Then, there exists universal constant $c > 0$ such that for any positive semi-definite matrix M and for every $t \geq 0$, we have

$$\mathbb{P}\left[|\mathbf{z}^T M\mathbf{z} - \mathbb{E}[\mathbf{z}^T M\mathbf{z}]| > t\right] \leq 2\exp\left\{-c\min\left\{\frac{t^2}{K^4||M||_{\mathsf{F}}^2}, \frac{t}{K^2||M||_{\mathsf{op}}}\right\}\right\}. \tag{4.30}$$

The next result bounds the eigenvalues of the $n\times n$ matrix $\mathbf{A} = \mathbf{\Phi}_{\mathsf{train}}\mathbf{\Phi}_{\mathsf{train}}^\top = \sum_{j=1}^d \lambda_j \mathbf{z}_j\mathbf{z}_j^\top$, where $\mathbf{z}_j$ are i.i.d. with $\mathbf{z}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$. Let $\mu_1(\mathbf{A})$ denote the largest eigenvalue and $\mu_n(\mathbf{A})$ denote the smallest eigenvalue of $\mathbf{A}$ respectively.

From [10][10], we have the following result which is a stronger version of the result from Lemma 1 from Chapter 3:

Lemma 20. *With probability at least* $(1 - 2e^{-n})$*, the eigenvalues of* $\mathbf{A}$ *satisfy:*

$$\sum_j \lambda_j - \Diamond \leq \mu_n(\mathbf{A}) \leq \mu_1(\mathbf{A}) \leq \sum_j \lambda_j + \Diamond, \tag{4.31}$$

where,

$$\Diamond = \frac{32}{9}\left(\lambda_1(1+\ln 9)n + \sqrt{(1+\ln 9)n\sum_j \lambda_j^2}\right). \tag{4.32}$$

Next, as stated previously in Lemma 14 we will use this result to obtain bounds on the eigenvalues of $\mathbf{A}^{-1}$ assuming that λ_j are such that $\Diamond \ll \sum_j \lambda_j$.[11]

[10]More precisely this lemma appeared in the first version of this work at `https://arxiv.org/pdf/1906.11300v1.pdf`. In subsequent versions the authors use a slightly weaker version of this result since it is sufficient for their purpose.

[11]Note that in the regime $q + r < 1$ (where regression works from 2 in Chapter 3), we do not have $\Diamond \ll \lambda_j$ and in such scenarios we cannot simply rely on eigenvalue bounds and need to use other techniques in the proof.

Lemma 14. ***(Eigenvalue bounds on $\mathbf{A}^{-1}$ adapted from [10])****:*
If $\mathbf{\Lambda}$ is such that $\lozenge \ll \sum_j \lambda_j$, then with probability at least $(1 - 2e^{-n})$,

$$\bar{\mu} - \Delta_\mu \leq \mu_n(\mathbf{A}^{-1}) \leq \mu_1(\mathbf{A}^{-1}) \leq \bar{\mu} + \Delta_\mu,$$

where,

$$\bar{\mu} = \frac{1}{\sum_j \lambda_j} \tag{4.22}$$

$$\lozenge = \frac{32}{9}\left(\lambda_1(1+\ln 9)n + \sqrt{(1+\ln 9)n\sum_j \lambda_j^2}\right) \tag{4.23}$$

$$\Delta_\mu = \bar{\mu}\left(\frac{\lozenge}{\sum_j \lambda j} + \Theta\left(\frac{\lozenge}{\sum_j \lambda j}\right)^2\right). \tag{4.24}$$

Further this implies that with probability at least $(1 - 2e^{-n})$,

$$\left|\mu_i(\mathbf{A}^{-1} - \bar{\mu}\mathbf{I}_n)\right| \leq \Delta_\mu$$

for all $i \in [n]$.

Proof. Let $S = \sum_j \lambda_j$.

$$\frac{1}{S+\lozenge} = \frac{1}{S}\left(1 + \frac{\lozenge}{S}\right)^{-1} \tag{4.33}$$

$$= \frac{1}{S}\left(1 - \frac{\lozenge}{S} + \Theta\left(\frac{\lozenge}{S}\right)^2\right) \tag{4.34}$$

$$= \bar{\mu} - \Delta_\mu, \tag{4.35}$$

and analogously $(S - \lozenge)^{-1} = \bar{\mu} + \Delta_\mu$. Taking reciprocals of everything in the inequality 4.31, and since the eigenvalues of $\mathbf{A}$ and $\mathbf{A}^{-1}$ are reciprocals of each other, the desired result follows.

□

As a Corollary of Lemma 14:

Corollary 5. ***(Asymptotic eigenvalue bounds on $\mathbf{A}^{-1}$)*** *Considering the asymptotic scaling of the model parameters from the bi-level model (Definition 9), in the regime $1 < q + r < (1+p)/2$,*

$$\bar{\mu} = n^{-p}$$
$$\Delta_\mu \leq c_4 n^{1-p-q-r} \ll \bar{\mu},$$

where $\bar{\mu}$ and Δ_μ are defined as in Lemma 14, and c_4 is a universal constant.

Proof. From the asymptotic scaling of the λ_j from (4.9) and (4.10), we see that (from the definition provided in Lemma 14)

$$\begin{aligned}\bar{\mu} &= \frac{1}{\sum_j \lambda_j} \\ &= \frac{1}{n^r n^{p-q-r} + (n^p - n^r)(1-n^q) \cdot n^p/(n^p - n^r))} \\ &= \frac{1}{n^{p-q} + n^p - n^{p-q}} \\ &= n^{-p}.\end{aligned}$$

Next, we have that

$$\begin{aligned}\Diamond &= \frac{32}{9}\left(\lambda_1(1+\ln 9)n + \sqrt{(1+\ln 9)n\sum_j \lambda_j^2}\right) \\ &\leq c_1 n^{1+p-q-r} + c_2\sqrt{n(n^r n^{2p-2q-2r} + (n^p - n^r))} \\ &\leq c_1 n^{1+p-q-r} + c_2\sqrt{n^{1+2p-2q-r} + n^{1+p}}\end{aligned}$$

for constants c_1 and c_2,

The second term is of the order $n^{\max((1-r)/2+p-q,(1+p)/2)}$. Thus, in the regime $q+r < (1+p)/2$, and since $r<1$ we have $1+p-q-r > (1-r)/2+p-q$ and $1+p-q-r > (1+p)/2$ and the first term dominates.

Thus, $\Diamond \leq c_3 n^{1+p-q-r}$ for some constant c_3 and sufficiently large n.

Observe that since $q+r>1$, $\Diamond \ll \sum_j \lambda_j = n^p$. Thus, we can substitute into our relation for Δ_μ from Lemma 14, to see that

$$\begin{aligned}\Delta_\mu &= \bar{\mu}\left(\frac{\Diamond}{\sum_j \lambda j} + \Theta\left(\frac{\Diamond}{\sum_j \lambda j}\right)^2\right) \\ &\leq n^{-p}\left((c_3 n^{1+p-q-r})(n^{-p}) + \Theta((c_3 n^{1+p-q-r})^2(n^{-p})^2)\right) \\ &= n^{-p}(c_3 n^{1-q-r} + \Theta(c_3 n^{2(1-q-r)})).\end{aligned}$$

In the regime where $q+r>1$, the first term in the sum dominates the second, giving us,

$$\Delta_\mu \leq c_4 n^{1-p-q-r}$$

for some constant c_4 and sufficiently large n. This completes the proof. □

Finally, in this section, we restate well-known bounds concerning Gaussian random variables.

Lemma 21. *Chi-squared tail bound:*
Let $\mathbf{z} \sim \mathcal{N}(0, I_n)$*. For any* $\delta \in (0,1)$*, with probability at least* $(1 - 2e^{-n\delta^2})$ *we have:*

$$n(1-\delta) \leq \|\mathbf{z}\|^2 \leq n(1+\delta). \tag{4.36}$$

From bounds on the expectation of the maximum of k Gaussians:

Lemma 22. *Let* $\mathbf{z}_\tau = \max_{1 \leq j \leq k} \mathbf{z}_j$ *where* $\mathbf{z}_j \sim \mathcal{N}(0,1)$*. Then,*

$$\frac{1}{\sqrt{\pi \ln 2}} \cdot \sqrt{\ln k} \leq \mathbb{E}[\mathbf{z}_\tau] \leq \sqrt{2} \cdot \sqrt{\ln k}. \tag{4.37}$$

4.10 Appendix: Utility bounds

The big technical challenge in moving from binary classification (as was studied in Chapter 3 to multiclass classification has to do with the nature of the training data. Whereas for binary classification one could change coordinates so that the binary labels only depended on a single Gaussian random variable and were independent of all other directions of Gaussian variation in the covariates, no such change of coordinates exists for multiclass labels. The one-hot-style encoding of the labels fundamentally depends on the realizations of all k of the Gaussian random variables representing each of the k classes. This means that we can no longer simply leverage independence to simplify the analysis and certain clever approaches used to invoke Hanson-Wright are no longer available to us. However, the need remains to appropriately bound quadratic forms of the form $\left|\mathbf{z}_j^\top \mathbf{A}^{-1} \boldsymbol{\Delta} y\right|$ both for the cases when j represents a feature that is not dominant in the computation of $\boldsymbol{\Delta} y$ as well as in cases where j represents a feature that is dominant in $\boldsymbol{\Delta} y$. To be able to control such quantities in the absence of the independence we could leverage in the binary case, this section derives two lemmas which can be viewed as helper bounds. These bounds will later be used to bound the various quantities from (4.15). Because our focus is on the asymptotic scaling, we will use c_i to denote the appropriate global constants.

In the subsequent lemmas, $\bar{\mu}$ and Δ_μ are defined as in the bounds on the eigenvalues of $\mathbf{A}^{-1}$ from Lemma 14. The following lemma is used to upper-bound the contamination term $\mathsf{CN}_{\tau,\zeta}$ in Lemma 18:

Lemma 23. *Let* $\boldsymbol{\Delta} y = \mathbf{y}_\tau - \mathbf{y}_\zeta$*. Let* τ*,* ζ*, and* j *be distinct. Then, with probability at least* $(1 - 7/(ndk))$*, we have,*

$$\left|\mathbf{z}_j^\top \mathbf{A}^{-1} \boldsymbol{\Delta} y\right| \leq c_7 (\bar{\mu} \sqrt{\frac{n}{k}} \cdot \sqrt{\ln(ndk)} + \Delta_\mu \cdot n/\sqrt{k}),$$

for some constant c_7*.*

This next lemma is used to bound the numerator of the survival variation term from (4.15):

Lemma 24. *Let $\boldsymbol{\Delta} y = \mathbf{y}_\tau - \mathbf{y}_\zeta$. With probability at least $(1 - 5/(nk))$, we have each of*

$$\mathbf{z}_\tau^\top \mathbf{A}^{-1} \boldsymbol{\Delta} y \leq \bar{\mu}(\mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\tau] - \mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\zeta]) + c_9(\bar{\mu}\sqrt{n}\sqrt{\ln(nk)} + \Delta_\mu \cdot n/\sqrt{k}) \tag{4.38}$$

$$\mathbf{z}_\tau^\top \mathbf{A}^{-1} \boldsymbol{\Delta} y \geq \bar{\mu}(\mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\tau] - \mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\zeta]) - c_9(\bar{\mu}\sqrt{n}\sqrt{\ln(nk)} + \Delta_\mu \cdot n/\sqrt{k}), \tag{4.39}$$

for some constant c_9.

The following corollary of the above is used to lower-bound the relative survival $\widehat{h}_{\tau,\zeta}[\tau]$, which in turn bounds the SU/CN ratio and the denominator of the survival variation term:

Corollary 6. *Let $\boldsymbol{\Delta} y = \mathbf{y}_\tau - \mathbf{y}_\zeta$. With probability at least $(1 - 5/(nk))$, we have,*

$$\mathbf{z}_\tau^\top \mathbf{A}^{-1} \boldsymbol{\Delta} y \geq c_{10}\bar{\mu}\frac{n}{k}\sqrt{\ln(k)} - c_9(\bar{\mu}\sqrt{n}\sqrt{\ln(nk)} + \Delta_\mu \cdot n/\sqrt{k}),$$

for some constant c_{10}.

Proof of Lemma 23

We will write $\mathbf{A}^{-1} = \bar{\mu}\mathbf{I}_n + \boldsymbol{\Delta} A_{inv}$, and split up the expression $\mathbf{z}_j^\top \mathbf{A}^{-1} \boldsymbol{\Delta} y$ into components involving $\bar{\mu}\mathbf{I}_n$, and components involving $\boldsymbol{\Delta} A_{inv}$. To bound the first term, we will use Hanson-Wright, and to bound the second we will use Cauchy-Schwartz. Throughout the proof, we rely on the concentration of the eigenvalues of $\mathbf{A}^{-1}$.

Next, we bound the first term (we set aside the constant $\bar{\mu}$ for now and deal with it later).

Bounds on $\mathbf{z}_j^T(\mathbf{y}_\tau - \mathbf{y}_\zeta)$

Throughout this section, let j be a feature index distinct from τ and ζ. Define the diagonal matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ with diagonal entries given by:

$$M_{ii} = \begin{cases} 1, & \text{if } \Delta y[i] \neq 0 \\ 0, & \text{otherwise} \end{cases}.$$

In other words, M_{ii} is 1 only if training point i belongs to class τ or ζ and is 0 otherwise. Thus for each $i \in [n]$, $M_{ii} \sim Bernoulli(2/k)$ and are independent of each other. We introduce this matrix $\mathbf{M}$ to ensure that our bound reflects the fact that most of the entries of $\boldsymbol{\Delta} y$ are 0. In particular $\Delta y[i] \neq 0$ only if point i belongs to class τ or ζ and only contains roughly $2n/k$ non-zero entries.[12] Note that we have by definition,

$$\mathbf{z}_j^T \boldsymbol{\Delta} y = \mathbf{z}_j^T \mathbf{M} \boldsymbol{\Delta} y.$$

Our strategy is to bound $\mathbf{z}_j^\top \mathbf{M} \boldsymbol{\Delta} y$ for every typical realization $\mathcal{M}$ of the random variable $\mathbf{M}$ using the Hanson-Wright inequality. Subsequently, we will apply these bounds with high probability over typical realizations of $\mathbf{M}$ that satisfy the Proposition below, which merely asserts that with high probability, the number of 1s in $\boldsymbol{\Delta} y$ is close to its expected value.

[12] An alternative bounding technique that first converted $\mathbf{z}_j^\top \boldsymbol{\Delta} y$ to a quadratic form and applied Hanson-Wright would be looser by a factor of $\sqrt{k}$ if we did not introduce $\mathbf{M}$.

Proposition 2. *For $\delta \in (0,1)$, with probability at least $(1 - 2e^{-\frac{2n\delta^2}{3k}})$, the trace of $\mathbf{M}$ is bounded as:*

$$(1-\delta)\frac{2n}{k} \leq \|\mathbf{\Delta} y\|_2^2 = \mathsf{tr}(\mathbf{M}) \leq (1+\delta)\frac{2n}{k}. \tag{4.40}$$

Proof. Note that $\mathsf{tr}(\mathbf{M})$ is the sum of n i.i.d Bernoulli random variables with mean $2/k$. The result follows by application of the Chernoff bound. □

Note that once we fix the realization $\mathcal{M}$, the distributions of $\mathbf{z}_j$ and $\mathbf{\Delta} y$ will now have to be conditioned on this realization and we need to deal with the modified distributions while applying the Hanson-Wright inequality. In particular, once we know that a feature was not the winning feature, it is no longer zero-mean.

Now,

$$\begin{aligned}
\mathbf{z}_j^T \mathcal{M} \mathbf{\Delta} y &= \sum_i z_j[i] \mathcal{M}_{ii} \Delta y[i] \\
&= \sum_{i:\mathcal{M}_{ii}=1} z_j[i] \Delta y[i] \\
&= \sum_{i:\mathcal{M}_{ii}=1} \left(z_j[i] - \mathbb{E}[z_j[i] \mid M_{ii}=1]\right) \Delta y[i] + \sum_{i:\mathcal{M}_{ii}=1} \mathbb{E}[z_j[i] \mid M_{ii}=1] \Delta y[i] \\
&= \sum_{i:\mathcal{M}_{ii}=1} \tilde{z}_{j,\mathcal{M}}[i] \Delta y[i] + \sum_{i:\mathcal{M}_{ii}=1} \mathbb{E}[z_j[i] \mid M_{ii}=1] \Delta y[i],
\end{aligned} \tag{4.41}$$

where $\tilde{z}_{j,\mathcal{M}}[i]$ is now a zero-mean random variable conditioned on the realization $\mathcal{M}$.

First, we bound the term $\sum_{i:\mathcal{M}_{ii}=1} \tilde{z}_{j,\mathcal{M}}[i] \Delta y[i]$. We collect the elements corresponding to indices where $\mathcal{M}_{ii} = 1$ into the vectors $\mathbf{z}'_{j,\mathcal{M}}$ and $\mathbf{\Delta} y'_{\mathcal{M}}$, which are both length $\mathsf{tr}(\mathcal{M})$ (Figure 4.7 shows an example of collecting elements).

$$\underbrace{\begin{bmatrix}1\\2\\3\\4\\5\end{bmatrix}}_{\tilde{\mathbf{z}}_{j,\mathcal{M}}}, \underbrace{\begin{bmatrix}1\\0\\-1\\1\\0\end{bmatrix}}_{\mathbf{\Delta} y} \rightarrow \underbrace{\begin{bmatrix}1\\3\\4\end{bmatrix}}_{\mathbf{z}'_{j,\mathcal{M}}}, \underbrace{\begin{bmatrix}1\\-1\\1\end{bmatrix}}_{\mathbf{\Delta} y'_{\mathcal{M}}}$$

Figure 4.7. An example of collecting elements at indices where $\mathcal{M}_{ii} = 1$ into smaller vectors of length $\mathsf{tr}(\mathcal{M})$. Recall that $\mathbf{\Delta} y[i] \neq 0$ iff $\mathcal{M}_{ii} = 0$.

We can then express

$$\begin{aligned}
&\sum_{i:\mathcal{M}_{ii}=1} \tilde{z}_{j,\mathcal{M}}[i]\Delta y[i] \\
&\quad = (\mathbf{z}'_{j,\mathcal{M}})^T \boldsymbol{\Delta} y'_{\mathcal{M}} \\
&\quad = \frac{1}{4}\left((\mathbf{z}'_{j,\mathcal{M}} + \boldsymbol{\Delta} y'_{\mathcal{M}})^T \mathbf{I}_{\mathsf{tr}(\mathcal{M})}(\mathbf{z}'_{j,\mathcal{M}} + \boldsymbol{\Delta} y'_{\mathcal{M}}) - (\mathbf{z}'_{j,\mathcal{M}} - \boldsymbol{\Delta} y'_{\mathcal{M}})^T \mathbf{I}_{\mathsf{tr}(\mathcal{M})}(\mathbf{z}'_{j,\mathcal{M}} - \boldsymbol{\Delta} y'_{\mathcal{M}})\right),
\end{aligned} \tag{4.42}$$

where we added and subtracted terms in the last equality.

We prove via the subsequent propositions that conditioned on the realization $\mathcal{M}$, the entries of $\mathbf{z}'_{j,\mathcal{M}} \pm \boldsymbol{\Delta} y'_{j,\mathcal{M}}$ are i.i.d. and sub-Gaussian with bounded norm. Thus, they satisfy the requirements to apply the Hanson-Wright inequality from [124] to bound the two quadratic forms in the above expression (4.42).

Proposition 3. *Conditioned on the realization $\mathcal{M}$, $z'_{j,\mathcal{M}}[i']$ has sub-Gaussian norm at most 6.*

Proof. Let i be the original index from which $z'_{j,\mathcal{M}}[i']$ was sampled.

If $j > k$, then $z'_{j,\mathcal{M}}[i'] = \tilde{z}_{j,\mathcal{M}}[i] = z_j[i]$ irrespective of the realization $\mathcal{M}$ because feature j is not used in the comparison to determine the class label and is independent to y_τ and y_ζ (and thus independent to $\mathbf{M}$). Further, $z_j[i]$ is simply a Gaussian (and therefore sub-Gaussian with sub-Gaussian norm $\|z_j[i]\|_{\psi_2} \leq 2$. Here we use the definition of sub-Gaussian norm from (4.29) reproduced here for convenience:

The sub-Gaussian norm of a random variable ξ is given by,

$$\begin{aligned}
\|\xi\|_{\psi_2} &= \inf_{K>0} K \\
&\quad \text{s.t. } \mathbb{E}\exp\left(\xi^2/K^2\right) \leq 2.
\end{aligned}$$

Otherwise, if j is one of the k features that define classes, since

$$\begin{aligned}
z'_{j,\mathcal{M}}[i'] &= \tilde{z}_{j,\mathcal{M}}[i] \\
&= z_j[i] - \mathbb{E}[z_j[i] \mid M_{ii} = 1],
\end{aligned}$$

the triangle inequality states that

$$\|\tilde{z}_{j,\mathcal{M}}[i]\|_{\psi_2} \leq \|z_j[i]\|_{\psi_2} + \|\mathbb{E}[z_j[i] \mid M_{ii} = 1]\|_{\psi_2}.$$

Note that the distribution of $z_j[i]$ conditioned on realization $\mathcal{M}$ is equivalent to the distribution obtained by conditioning on the event $M_{ii} = 1$. So it is sufficient to compute these sub-Gaussian norms conditioned on the event $M_{ii} = 1$.

We will first bound $\|z_j[i]\|_{\psi_2}$. Let $\mathcal{E}_j$ be the event that $z_j[i]$ is the maximum out of the first k features, and let $\mathcal{E}_j^c$ be the complementary event.

First, without conditioning on $\mathcal{E}_j$, we know by well-known results for the standard Gaussian that

$$\mathbb{E}\exp(\mathbf{z}_j[i]^2/5) = \sqrt{\frac{5}{3}} \leq \frac{4}{3}.$$

Using the law of iterated expectation we can relate this to the expectation conditioned on the events $\mathcal{E}_j$ and $\mathcal{E}_j^c$, noting that $\mathbb{P}(\mathcal{E}_j) = 1/k$:

$$\begin{aligned}
\frac{4}{3} &\geq \mathbb{E}\exp(\mathbf{z}_j[i]^2/5) \\
&= \mathbb{P}(\mathcal{E}_j)\mathbb{E}\exp(\mathbf{z}_j[i]^2/5|\mathcal{E}_j) + \mathbb{P}(\mathcal{E}_j^c)\mathbb{E}\exp(\mathbf{z}_j[i]^2/5|\mathcal{E}_j^c) \\
&= \frac{1}{k}\mathbb{E}\exp(\mathbf{z}_j[i]^2/5|\mathcal{E}_j) + \frac{k-1}{k}\mathbb{E}\exp(\mathbf{z}_j[i]^2/5|\mathcal{E}_j^c).
\end{aligned}$$

Rearranging terms, we obtain,

$$\begin{aligned}
\frac{k-1}{k}\mathbb{E}\exp(\mathbf{z}_j[i]^2/5|\mathcal{E}_j^c) &\leq \frac{4}{3} - \frac{1}{k}\mathbb{E}\exp(\mathbf{z}_j[i]^2/5|\mathcal{E}_j) \\
\implies \mathbb{E}\exp(\mathbf{z}_j[i]^2/5|\mathcal{E}_j^c) &\leq \frac{k}{k-1}\left(\frac{4}{3} - \frac{1}{k}\mathbb{E}\exp(\mathbf{z}_j[i]^2/5|\mathcal{E}_j)\right) \\
&\leq \frac{k}{k-1}\cdot\frac{4}{3} \\
&\leq 2,
\end{aligned}$$

where in the second to last inequality we used the non-negativity of $\mathbb{E}\exp(\mathbf{z}_j[i]^2/5|\mathcal{E}_j)$ and in the last equality we assumed $k \geq 3$. We then have

$$\begin{aligned}
\mathbb{E}\exp(\mathbf{z}_j[i]^2/5|\mathcal{E}_j^c) &= \sum_{m\neq j}\mathbb{E}\exp(\mathbf{z}_j[i]^2/5|\mathcal{E}_j^c \cap \mathcal{E}_m)\mathbb{P}(\mathcal{E}_m \mid \mathcal{E}_j^c) \\
&= \frac{1}{k-1}\sum_{m\neq j}\mathbb{E}\exp(\mathbf{z}_j[i]^2/5|\mathcal{E}_j^c \cap \mathcal{E}_m) \qquad (4.43)
\end{aligned}$$

where the last equality follows by symmetry. Further by symmetry, all the terms in the above summation that we are averaging are equal, so we can express it as an average of just the terms corresponding to $m = \tau$ and $m = \zeta$, as follows:

$$\begin{aligned}
(4.43) &= \frac{1}{2}\mathbb{E}\exp(\mathbf{z}_j[i]^2/5|\mathcal{E}_j^c \cap \mathcal{E}_\tau) + \frac{1}{2}\mathbb{E}\exp(\mathbf{z}_j[i]^2/5|\mathcal{E}_j^c \cap \mathcal{E}_\zeta) \\
&= \mathbb{P}(\mathcal{E}_\tau \mid \mathcal{E}_j^c \cap (\mathcal{E}_\tau \cup \mathcal{E}_\zeta))\mathbb{E}\exp(\mathbf{z}_j[i]^2/5|\mathcal{E}_j^c \cap \mathcal{E}_\tau) \\
&\quad + \mathbb{P}(\mathcal{E}_\zeta \mid \mathcal{E}_j^c \cap (\mathcal{E}_\tau \cup \mathcal{E}_\zeta))\mathbb{E}\exp(\mathbf{z}_j[i]^2/5|\mathcal{E}_j^c \cap \mathcal{E}_\zeta), \qquad (4.44)
\end{aligned}$$

again by symmetry. Since exactly one of $\mathcal{E}_\tau$ and $\mathcal{E}_\zeta$ are true when conditioned on $\mathcal{E}_j^c \cap (\mathcal{E}_\tau \cup \mathcal{E}_\zeta)$, we can rewrite the above as our desired expectation

$$\begin{aligned}(4.44) &= \mathbb{E}\exp(\mathbf{z}_j[i]^2/5 | \mathcal{E}_j^c \cap (\mathcal{E}_\tau \cup \mathcal{E}_\zeta)) \\ &= \mathbb{E}\exp(\mathbf{z}_j[i]^2/5 | \mathcal{E}_\tau \cup \mathcal{E}_\zeta) \\ &= \mathbb{E}\exp(\mathbf{z}_j[i]^2/5 | M_{ii} = 1),\end{aligned}$$

since $M_{ii} = 1$ is equivalent to the event $\mathcal{E}_\tau \cup \mathcal{E}_\zeta$. Thus, conditioned on the event $M_{ii} = 1$, $\|z_j[i]\|_{\psi_2} \leq \sqrt{5}$.

Next we consider $\|\mathbb{E}[z_j[i] \mid M_{ii} = 1]\|_{\psi_2}$. By a similar argument to above, we have that $\mathbb{E}[z_j[i] \mid M_{ii} = 1] = \mathbb{E}[z_j[i] \mid \mathcal{E}_j^c]$, so we will focus on the second quantity instead. Bounds on the max of Gaussians (Lemma 22) state that:

$$\begin{aligned} & & 0 < \mathbb{E}[\mathbf{z}_j[i] \mid \mathcal{E}_j] \leq \sqrt{2\log(k)} \\ \implies & & 0 > \mathbb{E}[\mathbf{z}_j[i] \mid \mathcal{E}_j^c] \geq -\frac{1}{k-1}\sqrt{2\log(k)} \geq -2 \\ \implies & & \exp\left(\frac{\mathbb{E}[\mathbf{z}_j[i] \mid \mathcal{E}_j^c]^2}{3^2}\right) < 2.\end{aligned}$$

In the second last inequality we use the fact that the function $f(k) = \left|\sqrt{2\log k}/(k-1)\right|$ is monotonically decreasing in k and assumed $k \geq 3$.

Thus, the (constant) random variable $\mathbb{E}[\mathbf{z}_j[i] \mid M_{ii} = 1]$ is sub-Gaussian with parameter 3. So, by the triangle inequality, conditioned on $M_{ii} = 1$

$$\begin{aligned}\|\tilde{z}_{j,m}[i]\|_{\psi_2} &\leq \|z_j[i]\|_{\psi_2} + \|\mathbb{E}[\tilde{z}_{j,m}\|_{\psi_2} \\ &\leq \sqrt{5} + 3 \\ &\leq 6.\end{aligned}$$

This completes the proof that conditioned on the realization $\mathcal{M}$, $z'_{j,\mathcal{M}}[i']$ is sub-Gaussian with norm at most 6. □

We can now prove our target result:

Proposition 4. *With probability at least* $(1 - 6/(ndk))$,

$$\left|\mathbf{z}_j^\top \boldsymbol{\Delta} y\right| \leq c_6 \sqrt{\frac{n}{k}} \cdot \sqrt{\log(ndk)}.$$

for universal constant c_6.

Proof. Our strategy will be to bound $\mathbf{z}_j^\top \boldsymbol{\Delta} y = \mathbf{z}_j^\top \mathbf{M} \boldsymbol{\Delta} y$ for every typical realization $\mathcal{M}$ of $\mathbf{M}$ that satisfies Proposition 2. Recall that for a given realization $\mathcal{M}$ we have,

$$\mathbf{z}_j^T \mathcal{M} \boldsymbol{\Delta} y = \sum_{i:\mathcal{M}_{ii}=1} \tilde{z}_{j,\mathcal{M}}[i] \Delta y[i] + \sum_{i:\mathcal{M}_{ii}=1} \mathbb{E}[z_j[i] \mid M_{ii} = 1] \Delta y[i]. \tag{4.45}$$

We will use Hanson-Wright to bound the first term, which we previously expressed in (4.42) as:

$$\sum_{i:\mathcal{M}_{ii}=1} \tilde{z}_{j,\mathcal{M}}[i]\Delta y[i]$$
$$= \frac{1}{4}\left((\mathbf{z}'_{j,\mathcal{M}} + \boldsymbol{\Delta} y'_{\mathcal{M}})^T \mathbf{I}_{\mathsf{tr}(\mathcal{M})}(\mathbf{z}'_{j,\mathcal{M}} + \boldsymbol{\Delta} y'_{\mathcal{M}}) - (\mathbf{z}'_{j,\mathcal{M}} - \boldsymbol{\Delta} y'_{\mathcal{M}})^T \mathbf{I}_{\mathsf{tr}(\mathcal{M})}(\mathbf{z}'_{j,\mathcal{M}} - \boldsymbol{\Delta} y'_{\mathcal{M}})\right).$$

By Proposition 3, the sub-Gaussian conditions for the entries of $\mathbf{z}'_{j,m}$ are satisfied. Further, $\boldsymbol{\Delta} y'_{\mathcal{M}}$ is bounded in $[-1, 1]$, so $\|\boldsymbol{\Delta} y'_{\mathcal{M}}\|_{\psi_2} \leq 2$. Thus, by the triangle inequality, the sub-Gaussian norm of the entries of $\mathbf{z}'_{j,\mathcal{M}} \pm \boldsymbol{\Delta} y'_{\mathcal{M}}$ is bounded by $K \leq 6 + 2 = 8$. Also note that conditioned on the realization $\mathcal{M}$, $\mathbf{z}'_{j,\mathcal{M}}$ is zero-mean by construction and $\boldsymbol{\Delta} y'_{\mathcal{M}}$ is zero-mean by symmetry between τ and ζ, so we can now apply the Hanson-Wright inequality to both terms.

We choose parameter

$$t = \frac{K^2}{\sqrt{c}}\sqrt{\mathsf{tr}(\mathcal{M})}\sqrt{\log(ndk)}.$$

where c is the constant from the Hanson-Wright result.

So

$$\frac{t^2}{K^4\|\mathbf{I}_{\mathsf{tr}(\mathcal{M})}\|_{\mathsf{F}}^2} = \frac{1}{c}\log(ndk)$$
$$\frac{t}{K^2\|\mathbf{I}_{\mathsf{tr}(\mathcal{M})}\|_{\mathsf{op}}} = \frac{1}{\sqrt{c}}\sqrt{\mathsf{tr}(\mathcal{M})}\sqrt{\log(ndk)} > \frac{1}{c}\log(ndk).$$

The last inequality follows since with high probability $\mathsf{tr}(\mathcal{M}) = \Theta(\sqrt{n/k})$, by Proposition 2, $\sqrt{\mathsf{tr}(\mathcal{M})}\sqrt{\log(ndk)} = \Theta(\sqrt{n\log(ndk)/k})$ grows faster than $\log(ndk)$.

Finally, note that:

$$\begin{aligned}
\mathbb{E}[(\mathbf{z}'_{j,\mathcal{M}})^T \boldsymbol{\Delta} y'_{\mathcal{M}} \mid \mathbf{M} = \mathcal{M}] &= \sum_{i:\mathcal{M}_{ii}=1} \mathbb{E}[\tilde{z}_{j,\mathcal{M}}[i]\Delta y[i] \mid \mathbf{M} = \mathcal{M}] \\
&= \sum_{i:\mathcal{M}_{ii}=1} \mathbb{E}[\tilde{z}_{j,\mathcal{M}}[i]\Delta y[i] \mid M_{ii} = 1] \\
&= \sum_{i:\mathcal{M}_{ii}=1} \frac{1}{2}\mathbb{E}[\tilde{z}_{j,\mathcal{M}}[i] \mid \Delta y[i] = 1] - \frac{1}{2}\mathbb{E}[\tilde{z}_{j,\mathcal{M}}[i] \mid \Delta y[i] = -1] \\
&= 0,
\end{aligned}$$

where the last equation follows by symmetry. Knowing which of $\mathbf{z}_\tau[i]$ or $\mathbf{z}_\zeta[i]$ was the maximum does not change the conditional expectation of $\tilde{z}_{j,\mathcal{M}}[i]$.

So, applying Hanson-Wright, with probability at least $(1 - 4/(ndk))$ we have

$$-\frac{K^2}{2}c_5\sqrt{\mathsf{tr}(\mathcal{M})}\sqrt{\log(ndk)} = -\frac{t}{2} \leq \tilde{\mathbf{z}}_{j,m}^T \Delta\mathbf{y} \leq \frac{t}{2} = \frac{K^2}{2}c_5\sqrt{\mathsf{tr}(\mathcal{M})}\sqrt{\log(ndk)},$$

where $c_5 = \frac{1}{\sqrt{c}}$.

We next consider the second term $\sum_{i:\mathcal{M}_{ii}=1} \mathbb{E}[z_j[i] \mid M_{ii} = 1]\Delta y[i]$ from (4.41) conditioned on the realization $\mathcal{M}$.

By an identical symmetry argument as for the previous term we have, $0 \geq \mathbb{E}[z_j[i] \mid \mathcal{E}_j^c] = \mathbb{E}[z_j[i] \mid M_{ii} = 1]$. Then as a consequence of Lemma 22 and using the fact that $M_{ii} = 1$ implies $z_j[i]$ is not the maximum of k Gaussians we have, $\mathbb{E}[z_j[i] \mid \mathcal{E}_j^c] \geq -2\sqrt{\log(k)}/(k-1)$. So we can bound

$$\left| \sum_{i:\mathcal{M}_{ii}=1} \mathbb{E}[z_j[i] \mid M_{ii} = 1]\Delta y[i] \right| \leq \frac{2\sqrt{\log(k)}}{k-1} \left| \sum_{i:\mathcal{M}_{ii}=1} \boldsymbol{\Delta} y_i \right| \leq \frac{2\delta'\sqrt{\log(k)}}{k-1}, \tag{4.46}$$

with probability $1 - 2e^{-\delta'^2/(6 \cdot \mathsf{tr}(\mathcal{M}))}$, by application of the Chernoff bound and using the fact that conditioned on $M_{ii} = 1$, $\Delta y[i]$ takes value ± 1 with probability half by symmetry among features τ and ζ.

Next, we apply the high probability bounds above on typical realizations $\mathcal{M}$. In particular, we substitute bounds on $\mathsf{tr}(\mathbf{M})$ from (4.40) from Proposition 2 with $\delta = 1/2$ into (4.46), and set $\delta' = \sqrt{6(1+\delta)(n/k)\log(ndk)}$. Then $e^{-\delta'^2/(6 \cdot \mathsf{tr}(\mathbf{M}))} \leq 1/(ndk)$ and $e^{-\frac{2n\delta^2}{3k}} < 1/(ndk)$, so using the union bound we have with probability at least $(1 - 4/(ndk) - 1/(ndk) - 1/(ndk))$,

$$\begin{aligned}
\left|\mathbf{z}_j^T \Delta \mathbf{y}\right| &\leq \left| \sum_{i:\mathcal{M}_{ii}=1} \tilde{z}_{j,\mathcal{M}}[i]\Delta y[i] \right| + \left| \sum_{i:\mathcal{M}_{ii}=1} \mathbb{E}[z_j[i] \mid M_{ii} = 1]\Delta y[i] \right| \\
&\leq \frac{K^2}{2} c_5 \sqrt{1+\delta} \cdot \sqrt{\frac{2n}{k}} \cdot \sqrt{\log(ndk)} + \frac{2\sqrt{(1+\delta)(n/k)\log(ndk)}\sqrt{\log(k)}}{k-1} \\
&\leq \frac{K^2}{2} c_5 \sqrt{1+\delta} \cdot \sqrt{\frac{2n}{k}} \cdot \sqrt{\log(ndk)} + \frac{2\sqrt{(1+\delta)}\sqrt{\log(k)}}{k-1} \cdot \sqrt{\frac{n}{k}} \cdot \sqrt{\log(ndk)} \\
&\leq c_6 \sqrt{\frac{n}{k}} \cdot \sqrt{\log(ndk)},
\end{aligned} \tag{4.47}$$

for a suitable choice of c_6. □

Bounds on $\mathbf{z}_j^\top \mathbf{A}^{-1}(\mathbf{y}_\tau - \mathbf{y}_\zeta)$

We can now prove bounds on our target quantity. We restate the lemma that we are trying to prove below for convenience.

Lemma 23. *Let $\boldsymbol{\Delta} y = \mathbf{y}_\tau - \mathbf{y}_\zeta$. Let τ, ζ, and j be distinct. Then, with probability at least $(1 - 7/(ndk))$, we have,*

$$\left|\mathbf{z}_j^\top \mathbf{A}^{-1} \boldsymbol{\Delta} y\right| \leq c_7(\bar{\mu}\sqrt{\frac{n}{k}} \cdot \sqrt{\ln(ndk)} + \Delta_\mu \cdot n/\sqrt{k}),$$

for some constant c_7.

Proof. We can rewrite

$$\begin{aligned}\mathbf{z}_j^\top \mathbf{A}^{-1}\boldsymbol{\Delta} y &= \mathbf{z}_j^\top \left(\bar{\mu}\mathbf{I}_n + \boldsymbol{\Delta} A_{inv}\right)\boldsymbol{\Delta} y \\ &= \bar{\mu}\mathbf{z}_j^\top \boldsymbol{\Delta} y + \mathbf{z}_j^\top \boldsymbol{\Delta} A_{inv}\boldsymbol{\Delta} y.\end{aligned}$$

Next we can bound $\left|\mathbf{z}_j^\top \boldsymbol{\Delta} A_{inv}\boldsymbol{\Delta} y\right|$ simply as

$$\begin{aligned}|\mathbf{z}_j^\top \boldsymbol{\Delta} A_{inv}\boldsymbol{\Delta} y| &\leq \|\mathbf{z}_j\|_2 \|\boldsymbol{\Delta} A_{inv}\boldsymbol{\Delta} y\|_2 \\ &\leq \|\boldsymbol{\Delta} A_{inv}\|_{op}\|\mathbf{z}_j\|_2\|\boldsymbol{\Delta} y\|_2 \\ &\leq \Delta_\mu \|\mathbf{z}_j\|_2 \|\boldsymbol{\Delta} y\|_2,\end{aligned} \tag{4.48}$$

where we use the fact that $\boldsymbol{\Delta} A_{inv}$ is a symmetric matrix and its 2-norm is its maximum absolute eigenvalue. We obtain the eigenvalue bounds for $\boldsymbol{\Delta} A_{inv}$ from Lemma 14, holding with probability at least $1 - 2e^{-n}$.

So, by the triangle inequality, we have with probability at least $(1 - 6/(ndk) - 2e^{-n} - 2e^{-\frac{2n\delta^2}{3k}} - 2e^{-n\delta^2})$

$$\left|\mathbf{z}_j^\top \mathbf{A}^{-1}\boldsymbol{\Delta} y\right| \leq c_6\bar{\mu}\sqrt{\frac{n}{k}} \cdot \sqrt{\ln(ndk)} + \Delta_\mu \cdot \sqrt{(1+\delta)n} \cdot \sqrt{(1+\delta)\frac{2n}{k}}.$$

The first term follows from Proposition 4, and the second from our bound on $\mathsf{tr}(\mathbf{M}) = \|\boldsymbol{\Delta} y\|_2^2$ from Proposition 2, as well as an analogous application of the chi-squared bound (Lemma 21) on $\|\mathbf{z}_j\|_2$.

The proof follows by setting δ to any value in $(0, 1)$, choosing an appropriate constant c_7, and noting that for large enough n, $1/(ndk) \gg d_1 e^{-d_2 n/k}$ for any positive constants d_1, d_2. □

Proof of Lemma 24

Next we use a similar technique as in Section 4.10 to bound $\mathbf{z}_\tau^\top \mathbf{A}^{-1}\boldsymbol{\Delta} y$. We will write $\mathbf{A}^{-1} = \bar{\mu}\mathbf{I}_n + \boldsymbol{\Delta} A_{inv}$, and split up the expression $\mathbf{z}_\tau^\top \mathbf{A}^{-1}\boldsymbol{\Delta} y$ into components involving $\bar{\mu}\mathbf{I}_n$, and components involving $\boldsymbol{\Delta} A_{inv}$.

Proposition 5. *Consider two arbitrary length-n zero-mean vectors $\mathbf{y}$ and $\mathbf{z}$ whose components each has sub-Gaussian norm at most K. With probability at least $1 - 4/(nk)$ we have each of*

$$\begin{aligned}\mathbf{z}^\top \mathbf{y} &\leq \mathbb{E}[\mathbf{z}^\top \mathbf{y}] + 2c_8\sqrt{n} \cdot \sqrt{\ln(nk)} \\ \mathbf{z}^\top \mathbf{y} &\geq \mathbb{E}[\mathbf{z}^\top \mathbf{y}] - 2c_8\sqrt{n} \cdot \sqrt{\ln(nk)},\end{aligned}$$

for some universal constant c_8.

Proof. The upper-bound follows as

$$\begin{aligned}\mathbf{z}^\top \mathbf{y} &= \frac{1}{4}\left((\mathbf{z}+\mathbf{y})^\top(\mathbf{z}+\mathbf{y}) - (\mathbf{z}-\mathbf{y})^\top(\mathbf{z}-\mathbf{y})\right) \\ &\leq \mathbb{E}[\mathbf{z}^\top \mathbf{y}] + \frac{K^2}{2\sqrt{c}}\sqrt{n}\cdot\sqrt{\ln nk},\end{aligned}$$

with probability at least $(1 - 4/(nk))$, where we apply the Hanson-Wright inequality to each of the quadratic terms with $t = \frac{K^2}{\sqrt{c}}\sqrt{n}\sqrt{\ln(nk)}$ and use the fact that, letting $\mathbf{M} = \mathbf{I}_n$, $\|\mathbf{M}\|_F^2 = n, \|\mathbf{M}\|_{op} = 1$. The lower-bound can be obtained analogously, and an appropriate choice of c_8 completes the proof. □

From this, we can now prove Lemma 24, restated below for convenience:

Lemma 24. *Let $\boldsymbol{\Delta} y = \mathbf{y}_\tau - \mathbf{y}_\zeta$. With probability at least $(1 - 5/(nk))$, we have each of*

$$\mathbf{z}_\tau^\top \mathbf{A}^{-1}\boldsymbol{\Delta} y \leq \bar{\mu}(\mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\tau] - \mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\zeta]) + c_9(\bar{\mu}\sqrt{n}\sqrt{\ln(nk)} + \Delta_\mu \cdot n/\sqrt{k}) \tag{4.38}$$

$$\mathbf{z}_\tau^\top \mathbf{A}^{-1}\boldsymbol{\Delta} y \geq \bar{\mu}(\mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\tau] - \mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\zeta]) - c_9(\bar{\mu}\sqrt{n}\sqrt{\ln(nk)} + \Delta_\mu \cdot n/\sqrt{k}), \tag{4.39}$$

for some constant c_9.

Proof. We have

$$\begin{aligned}\mathbf{z}_\tau^\top \mathbf{A}^{-1}(\mathbf{y}_\tau - \mathbf{y}_\zeta) &= \mathbf{z}_\tau^\top \left(\bar{\mu}\mathbf{I}_n + \boldsymbol{\Delta} A_{inv}\right)(\mathbf{y}_\tau - \mathbf{y}_\zeta) \\ &= \bar{\mu}\mathbf{z}_\tau^\top(\mathbf{y}_\tau - \mathbf{y}_\zeta) + \mathbf{z}_\tau^\top \boldsymbol{\Delta} A_{inv}(\mathbf{y}_\tau - \mathbf{y}_\zeta) \\ &= \bar{\mu}\mathbf{z}_\tau^\top(\mathbf{y}_\tau - \mathbf{y}_\zeta) + \mathbf{z}_\tau^\top \boldsymbol{\Delta} A_{inv}(\mathbf{y}_\tau^{oh} - \mathbf{y}_\zeta^{oh}).\end{aligned}$$

We again simply bound

$$\begin{aligned}|\mathbf{z}_\tau^\top \boldsymbol{\Delta} A_{inv}\boldsymbol{\Delta} y| &\leq \|\mathbf{z}_j\|_2 \|\boldsymbol{\Delta} A_{inv}\boldsymbol{\Delta} y\|_2 \\ &\leq \|\boldsymbol{\Delta} A_{inv}\|_{op}\|\mathbf{z}_\tau\|_2\|\boldsymbol{\Delta} y\|_2 \\ &\leq \Delta_\mu \|\mathbf{z}_\tau\|_2 \|\boldsymbol{\Delta} y\|_2 \\ &\leq \Delta_\mu \cdot \sqrt{(1+\delta)n}\cdot\sqrt{(1+\delta)\frac{2n}{k}} \\ &= \Delta_\mu(1+\delta)\sqrt{2}\frac{n}{\sqrt{k}},\end{aligned}$$

with probability $(1 - 2e^{-n\delta^2} - 2e^{-\frac{2n\delta^2}{3k}})$, using chi-squared bounds for $\mathbf{z}_\tau$ (Lemma 21) and Chernoff bounds for $\boldsymbol{\Delta} y$ (Proposition 2).

With probability $(1 - 2e^{-n\delta^2} - 2e^{-\frac{2n\delta^2}{3k}})$, we get each of

$$\begin{aligned}\mathbf{z}_\tau^T \mathbf{A}^{-1}(\mathbf{y}_\tau - \mathbf{y}_\zeta) &\leq \bar{\mu}\mathbf{z}_\tau^T(\mathbf{y}_\tau - \mathbf{y}_\zeta) + \Delta_\mu(1+\delta)\sqrt{2}\frac{n}{\sqrt{k}} \\ \mathbf{z}_\tau^T \mathbf{A}^{-1}(\mathbf{y}_\tau - \mathbf{y}_\zeta) &\geq \bar{\mu}\mathbf{z}_\tau^T(\mathbf{y}_\tau - \mathbf{y}_\zeta) - \Delta_\mu(1+\delta)\sqrt{2}\frac{n}{\sqrt{k}}.\end{aligned}$$

By applying Proposition 5 on the relevant terms, setting δ to be an arbitrary value in $(0,1)$, and choosing an appropriate constant c_9, we obtain with probability $(1-5/(nk))$ each of

$$\mathbf{z}_\tau^T \mathbf{A}^{-1}(\mathbf{y}_\tau - \mathbf{y}_\zeta) \le \bar{\mu}(\mathbb{E}[\mathbf{z}_\tau^T \mathbf{y}_\tau] - \mathbb{E}[\mathbf{z}_\tau^T \mathbf{y}_\zeta]) + c_9(\bar{\mu}\sqrt{n}\sqrt{\ln(nk)} + \Delta_\mu n/\sqrt{k})$$
$$\mathbf{z}_\tau^T \mathbf{A}^{-1}(\mathbf{y}_\tau - \mathbf{y}_\zeta) \ge \bar{\mu}(\mathbb{E}[\mathbf{z}_\tau^T \mathbf{y}_\tau] - \mathbb{E}[\mathbf{z}_\tau^T \mathbf{y}_\zeta]) - c_9(\bar{\mu}\sqrt{n}\sqrt{\ln(nk)} + \Delta_\mu n/\sqrt{k}).$$

The probability comes from the union bound $(1 - 2e^{-n\delta^2} - 2e^{-\frac{2n\delta^2}{3k}} - 4/(nk)) \ge 1 - 5/(nk)$ (for sufficiently large n). □

Proof of Corollary 6

We claim the following bound:

Proposition 6. *Bounds on* $\mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\tau]$.

$$\frac{1}{\sqrt{\pi \ln 2}} \cdot \frac{n}{k} \cdot \sqrt{\ln k} \le \mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\tau] \le \sqrt{2} \cdot \frac{n}{k} \cdot \sqrt{\ln k} \tag{4.49}$$

Proof.

$$\begin{aligned}
\mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\tau] &= \mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\tau^{oh}] - \mathbb{E}[\mathbf{z}_\tau^\top \frac{1}{c}\mathbf{1}] \\
&= \mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\tau^{oh}] \\
&= n\left(\mathbb{E}[z_{\tau,i} y_{\tau,i}^{oh} | y_{\tau,i}^{oh} = 1]\mathbb{P}(y_{\tau,i}^{oh} = 1) + \mathbb{E}[z_{\tau,i} y_{\tau,i}^{oh} | y_{\tau,i}^{oh} = 0]\mathbb{P}(y_{\tau,i}^{oh} = 0)\right) \\
&= \frac{n}{k}\mathbb{E}[z_{\tau,i} | y_{\tau,i}^{oh} = 1].
\end{aligned}$$

So the desired bound follows from the bounds in Lemma 22. □

We can obtain a similar bound for when $\zeta \neq \tau$:

Proposition 7. *Bounds on* $\mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\zeta]$.

$$-\sqrt{2} \cdot \frac{n}{k} \cdot \frac{1}{k-1} \cdot \sqrt{\ln k} \le \mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\zeta] \le -\frac{1}{\sqrt{\pi \ln 2}} \cdot \frac{n}{k} \cdot \frac{1}{k-1} \cdot \sqrt{\ln k} \tag{4.50}$$

Proof. Observe that,

$$\begin{aligned}
\mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\zeta] &= \mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\zeta^{oh}] - \mathbb{E}[\mathbf{z}_\tau^\top \frac{1}{k}\mathbf{1}] \\
&= \mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\zeta^{oh}] - \frac{1}{k}\mathbb{E}[\mathbf{z}_\tau]^\top \mathbf{1} \\
&= \mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\zeta^{oh}] \\
&= \sum_i \mathbb{E}[z_{\tau,i} y_{\zeta,i}^{oh}] \\
&= n\left(\mathbb{E}[z_{\tau,i} y_{\zeta,i}^{oh} | y_{\zeta,i}^{oh} = 1]\mathbb{P}(y_{\zeta,i}^{oh} = 1) + \mathbb{E}[z_{\tau,i} y_{\zeta,i}^{oh} | y_{\zeta,i}^{oh} = 0]\mathbb{P}(y_{\zeta,i}^{oh} = 1)\right) \\
&= \frac{n}{k}\mathbb{E}[z_{\tau,i} | y_{\zeta,i}^{oh} = 1] \qquad (4.51)
\end{aligned}$$

Now, observe that

$$\begin{aligned}
\mathbb{E}[z_{\tau,i} | y_{\tau,i}^{oh} = 0] &= \sum_{\zeta \neq \tau} \mathbb{E}[z_{\tau,i} | y_{\zeta,i}^{oh} = 1]\mathbb{P}(y_{\zeta,i} = 1 \mid y_{\tau,i}^{oh} = 0) \\
&= \frac{1}{k-1}\sum_{\zeta \neq \tau} \mathbb{E}[z_{\tau,i} | y_{\zeta,i}^{oh} = 1] \\
&= \mathbb{E}[z_{\tau,i} | y_{\zeta,i}^{oh} = 1]
\end{aligned}$$

for a particular ζ, by symmetry over the possible ζ.

Next we bound $\mathbb{E}[z_{\tau,i} | y_{\tau,i}^{oh} = 0]$ as follows:

$$\begin{aligned}
& \mathbb{E}[z_{\tau,i} | y_{\tau,i}^{oh} = 1]\mathbb{P}(y_{\tau,i}^{oh} = 1) \\
+ \quad & \mathbb{E}[z_{\tau,i} | y_{\tau,i}^{oh} = 0]\mathbb{P}(y_{\tau,i}^{oh} = 0) = \mathbb{E}[z_{\tau,i}] = 0 \\
\implies \quad & \mathbb{E}[z_{\tau,i} | y_{\tau,i}^{oh} = 0]\frac{k-1}{k} = -\mathbb{E}[z_{\tau,i} | y_{\tau,i}^{oh} = 1]\frac{1}{k} \\
\implies \quad & \mathbb{E}[z_{\tau,i} | y_{\tau,i}^{oh} = 0] = -\mathbb{E}[z_{\tau,i} | y_{\tau,i}^{oh} = 1]\frac{1}{k-1}
\end{aligned}$$

Thus, substituting in the results from Lemma 22, and plugging back into (4.51), we obtain

$$-\sqrt{2} \cdot \frac{n}{k} \cdot \frac{1}{k-1} \cdot \sqrt{\ln k} \leq \mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\zeta] \leq -\frac{1}{\sqrt{\pi \ln 2}} \cdot \frac{n}{k} \cdot \frac{1}{k-1} \cdot \sqrt{\ln k}, \qquad (4.52)$$

the desired result. □

We can now prove Corollary 6, which we restate below for convenience:

Corollary 6. *Let $\boldsymbol{\Delta} y = \mathbf{y}_\tau - \mathbf{y}_\zeta$. With probability at least $(1 - 5/(nk))$, we have,*

$$\mathbf{z}_\tau^\top \mathbf{A}^{-1} \boldsymbol{\Delta} y \geq c_{10} \bar{\mu} \frac{n}{k} \sqrt{\ln(k)} - c_9(\bar{\mu}\sqrt{n}\sqrt{\ln(nk)} + \Delta_\mu \cdot n/\sqrt{k}),$$

for some constant c_{10}.

Proof. This follows by substituting the lower bound from (4.49) in Proposition 6 and the upper bound from (4.50) in Proposition 7 into (4.39) from Lemma 24, making an appropriate choice for c_{10}. □

4.11 Appendix: Misclassification events: Proof of Lemmas used in Theorem 5

With the previous section's utility bounds that allow us to deal with multiclass training data in hand, we are in a position to establish all the lemmas that we need to analyze misclassification.

Proof of Lemma 15: Lower bound on $\min_\zeta(Z[\tau] - Z[\zeta])$

With these bounds in hand, we can look at each misclassification event in turn. The first event to consider is if the best competing feature is unusually close to the true (maximum) feature.

Lemma 15. *(**Lower bound on the closest feature margin as** $k \to \infty$): For any constant $\varepsilon > 0$, there exists a constant θ such that, for sufficiently large k with probability at least $(1 - \varepsilon)$,*

$$\min_{\zeta:1\leq\zeta\neq\tau\leq k} (Z[\tau] - Z[\zeta]) \geq \frac{\theta}{\sqrt{2\ln(k)}}.$$

Here, τ is fixed and corresponds to the index of the true class — i.e. τ corresponds to the index of the maximum feature among the first k features.

Proof. The following result from [1] (reproduced in [133]) enables us to bound the closest feature margin as:

$$\mathbb{P}\left(\min_{\zeta} (Z[\tau] - Z[\zeta]) > \frac{\theta}{\sqrt{2\ln(k)}}\right) \geq c_{11}e^{-\theta},$$

for some universal positive constant c_{11}, for sufficiently large k. Thus, by selecting a constant θ such that $c_{11}e^{-\theta} = 1 - \varepsilon$ and choosing a sufficiently large k, we have that with probability $(1 - \varepsilon)$:

$$\min_{\zeta}(Z[\tau] - Z[\zeta]) \geq \frac{\theta}{\sqrt{2\ln k}}.$$

□

Lemma 16. *(**Lower bound on the closest feature margin when** k **is constant**): If $k = c_k$ for some fixed constant c_k, for any constant $\varepsilon > 0$, there exists a constant $\varepsilon' > 0$ such that*

$$\mathbb{P}\left(\min_{\zeta,\gamma:1\leq\zeta\neq\gamma\leq c_k} |Z[\zeta] - Z[\gamma]| \geq \varepsilon'\right) \geq 1 - \varepsilon.$$

Thus, with probability at least $(1-\varepsilon)$,

$$\min_{\zeta:1\leq\zeta\neq\tau\leq k} (Z[\tau] - Z[\zeta]) \geq \varepsilon'.$$

Here, τ is fixed and corresponds to the index of the true class — i.e. τ corresponds to the index of the maximum feature among the first k features.

Proof. Observe that,

$$\min_{1\leq\zeta\neq\tau\leq c_k} (Z[\tau] - Z[\zeta]) \geq \min_{1\leq\zeta\neq\gamma\leq c_k} |Z[\gamma] - Z[\zeta]|.$$

In other words, rather than bounding the margin between the largest and second-largest features, we will lower-bound the absolute difference between any pair of features.

Consider a particular (ζ, γ) tuple. Observe that $Z[\zeta] - Z[\gamma] \sim N(0, 2)$, since each feature is drawn independently from a standard Gaussian. For any $\epsilon' > 0$, we can upper-bound

$$\mathbb{P}\left(|Z[\zeta] - Z[\gamma]| \leq \varepsilon'\right) \leq \frac{\varepsilon'}{\sqrt{\pi}}$$

by taking the product of the maximum value of the Gaussian pdf and the width, $2\epsilon'$, of the region we are interested in. Taking the union bound across all (ζ, γ) tuples, we find that

$$\mathbb{P}\left(\min_{1\leq\zeta\neq\gamma\leq c_k} |Z[\zeta] - Z[\gamma]| \leq \varepsilon'\right) \leq \frac{c_k^2\varepsilon'}{\sqrt{\pi}}.$$

So for any given $\varepsilon > 0$, we can choose $\varepsilon' = \varepsilon\sqrt{\pi}/c_k^2$, and have that

$$\mathbb{P}\left(\min_{1\leq\zeta\neq\gamma\leq c_k} |Z[\zeta] - Z[\gamma]| \geq \varepsilon'\right) \geq 1 - \varepsilon.$$

□

Lower bound on $\frac{\lambda\widehat{h}_{\tau,\zeta}[\tau]}{\max_\zeta \mathsf{CN}_{\tau,\zeta}}$

Next, we will find a lower bound for survival-contamination ratio within the regime with low survival variance.

Lemma 17. *(**Lower bound on relative survival of true feature**): For any fixed $\zeta \in [k]$, $\zeta \neq \tau$, with $\lambda_\tau = \lambda_\zeta = \lambda$ we have with probability at least $(1 - 5/(nk))$,*

$$\lambda \widehat{h}_{\tau,\zeta}[\tau] \geq \lambda \left(c_{10} \bar{\mu} \frac{n}{k} \sqrt{\ln(k)} - c_9 (\bar{\mu} \sqrt{n} \sqrt{\ln(nk)} + \Delta_\mu \cdot n/\sqrt{k}) \right),$$

for universal positive constants c_9 and c_{10}.

Proof. Using Corollary 6, we lower bound $\widehat{h}_{\tau,\zeta}[\tau]$ with probability at least $(1 - 5/(nk))$ as

$$\begin{aligned}
\widehat{h}_{\tau,\zeta}[\tau] &= \lambda_\tau^{-1/2} (\widehat{\alpha}_\tau[\tau] - \widehat{\alpha}_\zeta[\tau]) \\
&= \mathbf{z}_\tau^\top \mathbf{A}^{-1} \mathbf{y}_\tau - \mathbf{z}_\tau^\top \mathbf{A}^{-1} \mathbf{y}_\zeta \\
&\geq c_{10} \bar{\mu} \frac{n}{k} \sqrt{\ln(k)} - c_9 (\bar{\mu} \sqrt{n} \sqrt{\ln(nk)} + \Delta_\mu \cdot n/\sqrt{k}).
\end{aligned}$$

Multiplying through by λ gives the desired result. □

From the above result, under the scalings of our bi-level model we obtain:

Corollary 2. *Under the bi-level ensemble model 9, for any fixed $\zeta \in [k]$, $\zeta \neq \tau$, $\lambda_\tau = \lambda_\zeta = \lambda$ if $t < 1/2$, $t < 2(q + r - 1)$ and $1 < q + r < (p + 1)/2$, with probability at least $(1 - 5/(nk))$,*

$$\lambda \widehat{h}_{\tau,\zeta}[\tau] \geq c_{12} n^{1-q-r-t} \sqrt{\ln(k)},$$

for universal positive constant c_{12}.

Proof. Substituting our asymptotic scalings into the results from Lemma 17 and using the decay rate of $\bar{\mu} \asymp n^{-p}$ from Corollary 5 (which we can do since $1 < q + r < (p + 1)/2$), we find that

$$\begin{aligned}
\lambda \widehat{h}_{\tau,\zeta}[\tau] &\geq n^{p-q-r} \left(c_{10} \bar{\mu} \frac{n}{k} \sqrt{\ln(k)} - c_9 (\bar{\mu} \sqrt{n} \sqrt{\ln(nk)} + \Delta_\mu \cdot n/\sqrt{k}) \right) \\
&= c_{10} n^{1-q-r-t} \sqrt{\ln(k)} - c_9 n^{1/2-q-r} \sqrt{\ln(nk)} - c_9 n^{2-2q-2r-t/2} \\
&\geq c_{12} n^{\max(1-q-r-t, 2-2q-2r-t/2)} \sqrt{\ln(k)} \\
&= c_{12} n^{1-q-r-t+\max(0, 1-q-r+t/2)} \sqrt{\ln(k)} \\
&\geq c_{12} n^{1-q-r-t} \sqrt{\ln(k)},
\end{aligned}$$

for an appropriately chosen universal constant c_{12} and sufficiently large n. □

Next we upper bound $\max_\zeta \mathsf{CN}_{\tau,\zeta}$.

Lemma 18. *(**Upper bound on contamination**): For any fixed $\zeta \in [k]$, $\zeta \neq \tau$, with probability at least $(1 - 7/(nk))$,*

$$\mathsf{CN}_{\tau,\zeta} \leq c_7 \left(\bar{\mu} \sqrt{\frac{n}{k}} \cdot \sqrt{\ln(ndk)} + \Delta_\mu \cdot \frac{n}{\sqrt{k}} \right) \cdot \sqrt{\sum \lambda_j^2},$$

for universal positive constant c_7.

Proof. For each ζ we have,

$$\mathsf{CN}_{\tau,\zeta} = \sqrt{\left(\sum_{j \notin \{\tau,\zeta\}} \lambda_j^2 (\widehat{h}_{\zeta,\tau}[j])^2\right)}$$

For $j \notin \{\tau, \zeta\}$, by Lemma 23,

$$\begin{aligned}
\left|\widehat{h}_{\zeta,\tau}[j]\right| &= \left|\widehat{h}_{\tau,\zeta}[j]\right| \\
&= |\widehat{\alpha}_j - \widehat{g}_j| \\
&= \left|\mathbf{z}_j^\top \mathbf{A}^{-1}\mathbf{y}_\tau - \mathbf{z}_j^\top \mathbf{A}^{-1}\mathbf{y}_\zeta\right| \\
&= \left|\mathbf{z}_j^\top \mathbf{A}^{-1}(\mathbf{y}_\tau - \mathbf{y}_\zeta)\right| \\
&\leq c_7 \left(\bar{\mu}\sqrt{\frac{n}{k}} \cdot \sqrt{\ln(ndk)} + \Delta_\mu \cdot \frac{n}{\sqrt{k}}\right),
\end{aligned}$$

with probability $1 - 7/(ndk)$.

So taking the union bound over all $d-2$ terms in the expression for the contamination, we can upper-bound it as

$$\mathsf{CN}_{\tau,\zeta} \leq c_7 \left(\bar{\mu}\sqrt{\frac{n}{k}} \cdot \sqrt{\ln(ndk)} + \Delta_\mu \cdot \frac{n}{\sqrt{k}}\right) \cdot \sqrt{\sum \lambda_j^2},$$

with probability $(1 - 7/(nk))$, the desired result. □

Corollary 3. *Under the bi-level model 9, in the regime $1 < q+r < (p+1)/2$, with probability at least $(1 - 7/(nk))$,*

$$\mathsf{CN}_{\tau,\zeta} \leq c_{13} n^{(1-t-p)/2+\max(0,3/2-q-r)+\max(0,p/2-q-r/2)} \sqrt{\ln(ndk)},$$

for universal positive constant c_{13}.

Proof. Since $1 < q + r < (p+1)/2$, we can apply Corollary 5 to the result from Lemma 18 and substitute in the known scalings of various terms, to obtain

$$\begin{aligned}
\mathsf{CN}_{\tau,\zeta} &\leq c_7 (n^{1/2-t/2-p}\sqrt{\ln(ndk)} + c_4 n^{2-p-q-r-t/2})(n^{p-q-r/2} + n^{p/2}) \\
&\leq c_{13} n^{(1-t-p)/2+\max(0,3/2-q-r)+\max(0,p/2-q-r/2)} \sqrt{\ln(ndk)},
\end{aligned}$$

for an appropriately chosen universal positive constant c_{13}.

□

Proof of Lemma 19: Bounds on Survival Variance

Finally, we look at the error event where a competing feature has unusually high survival relative to the true feature, so it is incorrectly selected.

Lemma 19. *(**Upper bound on survival variance**): For any fixed competing feature $\zeta \in [k]$, $\zeta \neq \tau$ with $\lambda_\tau = \lambda_\zeta$, we have with probability at least $(1 - 15/(nk))$,*

$$\frac{\widehat{h}_{\tau,\zeta}[\tau] - \widehat{h}_{\zeta,\tau}[\zeta]}{\widehat{h}_{\tau,\zeta}[\tau]} \leq \frac{2c_9(\bar{\mu}\sqrt{n}\sqrt{\ln(nk)} + \Delta_\mu \cdot n/\sqrt{k})}{c_{10}\bar{\mu}\frac{n}{k}\sqrt{\ln(k)} - c_9(\bar{\mu}\sqrt{n}\sqrt{\ln(nk)} + \Delta_\mu \cdot n/\sqrt{k})}, \tag{4.25}$$

for universal positive constants c_9 and c_{10}.

Proof. We first consider the numerator of the LHS of (4.25). By Lemma 24, with probability at least $(1 - 5/(nk))$,

$$\begin{aligned}
\widehat{h}_{\tau,\zeta}[\tau] &= \lambda_\tau^{-1/2}(\widehat{\alpha}_\tau[\tau] - \widehat{\alpha}_\zeta[\tau]) \\
&= \mathbf{z}_\tau^\top \mathbf{A}^{-1}\mathbf{y}_\tau - \mathbf{z}_\tau^\top \mathbf{A}^{-1}\mathbf{y}_\zeta \\
&\leq \bar{\mu}(\mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\tau] - \mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\zeta]) + c_9(\bar{\mu}\sqrt{n}\sqrt{\ln(nk)} + \Delta_\mu \cdot n/\sqrt{k}).
\end{aligned}$$

Similarly, with probability at least $(1 - 5/(nk))$,

$$\begin{aligned}
\widehat{h}_{\zeta,\tau}[\zeta] &= \lambda_\zeta^{-1/2}(\widehat{\alpha}_\zeta[\zeta] - \widehat{\alpha}_\tau[\zeta]) \\
&= \mathbf{z}_\zeta^\top \mathbf{A}^{-1}\mathbf{y}_\zeta - \mathbf{z}_\zeta^\top \mathbf{A}^{-1}\mathbf{y}_\tau \\
&\geq \bar{\mu}(\mathbb{E}[\mathbf{z}_\zeta^\top \mathbf{y}_\zeta] - \mathbb{E}[\mathbf{z}_\zeta^\top \mathbf{y}_\tau]) - c_9(\bar{\mu}\sqrt{n}\sqrt{\ln(nk)} + \Delta_\mu \cdot n/\sqrt{k}).
\end{aligned}$$

By symmetry,

$$\begin{aligned}
\mathbb{E}[\mathbf{z}_\zeta^\top \mathbf{y}_\zeta] &= \mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\tau] \\
\mathbb{E}[\mathbf{z}_\zeta^\top \mathbf{y}_\tau] &= \mathbb{E}[\mathbf{z}_\tau^\top \mathbf{y}_\zeta].
\end{aligned}$$

Thus with probability at least $(1 - 10/(nk))$,

$$\widehat{h}_{\tau,\zeta}[\tau] - \widehat{h}_{\zeta,\tau}[\zeta] \leq 2c_9(\bar{\mu}\sqrt{n}\sqrt{\ln(nk)} + \Delta_\mu \cdot n/\sqrt{k}).$$

Using Corollary 6 to lower-bound the denominator of the LHS of (4.25), we obtain with probability at least $(1 - 15/(nk))$

$$\frac{\widehat{h}_{\tau,\zeta}[\tau] - \widehat{h}_{\zeta,\tau}[\zeta]}{\widehat{h}_{\tau,\zeta}[\tau]} \leq \frac{2c_9(\bar{\mu}\sqrt{n}\sqrt{\ln(nk)} + \Delta_\mu \cdot n/\sqrt{k})}{c_{10}\bar{\mu}\frac{n}{k}\sqrt{\ln(k)} - c_9(\bar{\mu}\sqrt{n}\sqrt{\ln(nk)} + \Delta_\mu \cdot n/\sqrt{k})}.$$

□

We can apply Corollary 5 to simplify our results from Lemma 19 in the asymptotic regime for the bi-level model.

Corollary 4. *Under the bi-level ensemble model 9, for any fixed $\zeta \in [k]$, $\zeta \neq \tau$, if $t < 1/2$, $t < 2(q + r - 1)$, and $1 < q + r < (p + 1)/2$, with probability at least $(1 - 15/(nk))$,*

$$\frac{\widehat{h}_{\tau,\zeta}[\tau] - \widehat{h}_{\zeta,\tau}[\zeta]}{\widehat{h}_{\tau,\zeta}[\tau]} < n^{-u},$$

for large enough n for some fixed $u > 0$.

Proof. Substituting, using Corollary 5, in the regime where $1 < q+r < (p+1)/2$ and $t < 1/2$, we find that

$$\begin{aligned}
\frac{\widehat{h}_{\tau,\zeta}[\tau] - \widehat{h}_{\zeta,\tau}[\zeta]}{\widehat{h}_{\tau,\zeta}[\tau]} &\leq \frac{2c_9(n^{1/2-p}\sqrt{\ln(nk)} + c_4 n^{2-p-q-r-t/2})}{c_{10}n^{1-p-t}\sqrt{\ln(k)} - c_9(n^{1/2-p}\sqrt{\ln(nk)} + c_4 n^{2-p-q-r-t/2})} \\
&\leq \frac{2c_9}{c_{10}} \cdot \frac{n^{1/2}\sqrt{\ln(nk)} + c_4 n^{2-q-r-t/2}}{n^{1-t} - (c_9/c_{10})n^{1/2}\sqrt{\ln(n)} + (c_9 \cdot c_4/c_{10})n^{2-q-r-t/2}} \\
&\leq c_{14}\frac{n^{1/2}\sqrt{\ln(nk)} + n^{2-q-r-t/2}}{n^{1-t}} \\
&\leq c_{14} n^{\max(t-1/2, t/2+1-q-r)}\sqrt{\ln(nk)},
\end{aligned}$$

for sufficiently large n and an appropriate choice of positive constant c_{14}. Thus, if $\max(t - 1/2, t/2 + 1 - q - r) < 0$, our quantity of interest tends to zero at a polynomial rate as $n \to \infty$, completing the proof. □

Chapter 5

Learning control using neural networks

5.1 Introduction

In Chapters 2, 3 and 4 we saw via the analysis of overparameterized generalized linear models that there is a distinction between what can be learned and what is actually learned, i.e. even if the model has the capacity to learn the right solution, the implicit bias induced by the model structure and training algorithm is critical for steering the model towards the right solution. In the case of our work, the implicit bias was due to the covariance matrix of our data favoring a few important directions and the use of gradient descent to learn minimum-norm interpolating solutions. But is this principle restricted to only overparameterized linear models or applicable more widely? To answer this question, in this chapter we investigate empirically the importance of implicit bias when learning *non-linear* neural-network models. We focus our attention on two control problems where purely linear solutions are known to be sub-optimal and it is unclear what sort of features must be used for generalized linear solutions. The first is the famous Witsenhausen counterexample, a 2-step finite horizon control problem, and the second is an infinite horizon problem of stabilizing a control system with multiplicative observation noise. In both these problems, we will see how intelligently choosing an architecture and training method by leveraging knowledge about the problem domain can help us more easily and robustly find good solutions.

5.2 Witsenhausen problem

Problem setup

The Witsenhausen problem is a a simple decentralized stochastic control problem with two controllers as illustrated in Figure 1.8 in Section 1.3, reproduced here for the convenience of reader. The first controller receives X_0 as input where X_0 is a zero-centered Gaussian random variable with variance σ_x^2. Observing X_0 perfectly, the first controller determines the control U_1 and the state evolves to be $X_1 = X_0 + U_1$. The second controller then receives

a noisy version of the state, $Y_2 = X_1 + Z$, where Z is a standard unit variance normal random variable. Given Y_2, the second controller determines the control U_2 and the final state evolves to be $X_2 = X_1 - U_2$. The controllers are designed together to ideally minimize the expected cost function $k^2\mathbb{E}[\|U_1\|^2] + \mathbb{E}[\|X_2\|^2]$. The two parameters σ_x^2 (measure of the

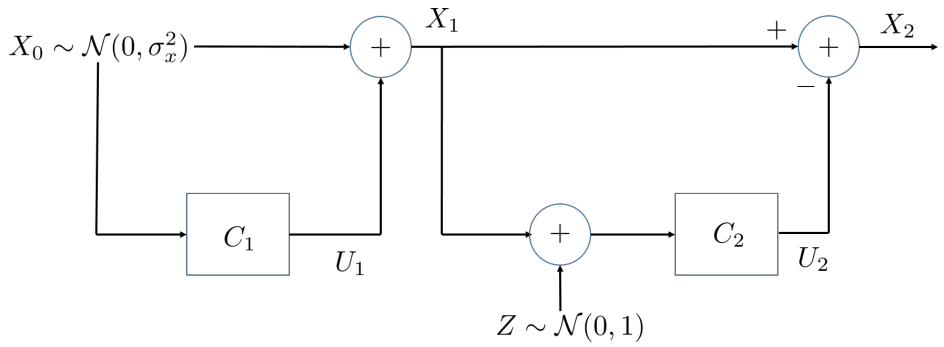


Figure 1.8. The Witsenhausen problem. The objective is to minimize $k^2\mathbb{E}[\|U_1\|^2] + \mathbb{E}[\|X_2\|^2]$. (repeated from page 10)

uncertainty in the initial state) and k^2 (how heavily we penalize the first controller's control input) define an instance of the Witsenhausen problem.

Challenges while designing controllers

First, we note that since $\mathbb{E}[\|X_2\|^2] = \mathbb{E}[\|X_1 - U_2\|^2]$ and U_2 is determined solely based on Y_2, the optimal control strategy for the second controller is to output the conditional expectation of X_1 given Y_2, $\mathbb{E}[X_1 \mid Y_2]$. However, the optimal control strategy for the first controller is more challenging to determine since the choice of U_1 influences the distribution of X_1 (and consequently the distribution of Y_2, U_2 and X_2) and it is unclear what choice will minimize the cost function, $k^2\mathbb{E}[\|U_1\|^2] + \mathbb{E}[\|X_2\|^2]$.

An interesting feature about the the Witsenhausen problem is that linear control strategies are provably sub-optimal [150]. The community has performed numerical explorations of non-linear strategies for the Witsenhausen problem [8] and in [55] it was observed that these non-linear strategies that visually resembled "slopey quantizers".

In our work, we investigate whether we can use neural-networks to learn non-linear control strategies for the Witsenhausen problem. We leverage the contemporary development of libraries and computational platforms designed to facilitate deep learning research [80] and use PyTorch [115] to train our neural networks.

First, we tried to replicate the results obtained by [8] using the architecture they used in 2001. Here, the first controller has one hidden layer of 150 units with sigmoid activations and the second controller has one hidden layer of 30 units with sigmoid activations. However, our results were extremely sensitive to the random seed used to generate X_0 and Z while training as well as the initialization of the parameters of the networks. Subfigure (b) in Figure 5.1 plots a histogram for the number of seeds that lead to a learned strategy with a

particular loss when $k^2 = 0.04$ and we see that only a small fraction of seeds achieve a low loss (close to 0.2). Subfigure (c) plots a similar histogram for $k^2 = 0.15$. It is a wonder that Baglietto et al.[8] were able to learn a good strategy with the computation power available in 2001 which was a minuscule fraction of what we have access to today.

Subfigure (d) visualizes the strategy for the first controller that led to the minimum loss of 0.20 by plotting the relationship between X_1 and X_0 when using the controller. Subfigures (e) and (f) visualize examples of strategies for the first controller that led to a higher loss value. We believe that part of the difficulty in learning a good strategy comes from the difficulty in escaping local minimas that exist due to the coupling between the first and second controller.

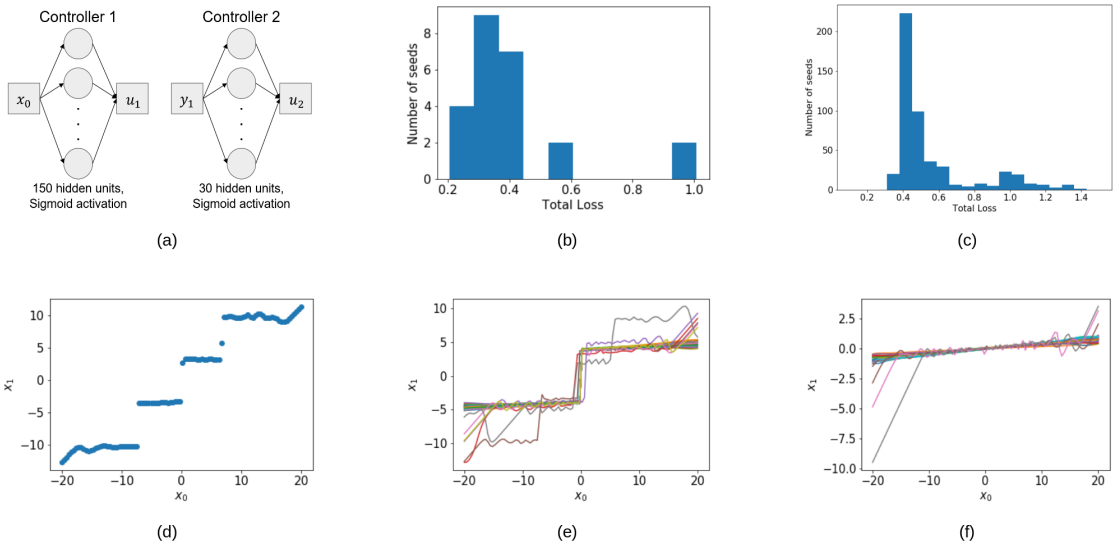


Figure 5.1. (a) Architecture used by Baglietto et.al. where controller 1 has 150 units of sigmoid activation and controller 2 has 30 units of sigmoid activation. (b) Histogram of total losses of 100 seeds with $k^2 = 0.04$, $\sigma_X = 5$, train batch size=10k, test batch size=10k, iterations=100k, learning rate=0.2. (c) Histogram of total losses of 500 seeds with $k^2 = 0.15$, $\sigma_X = 5$, train batch size=128, test batch size=10k, iterations=30k, learning rate=0.02. (d) Minimum loss strategy found using architecture described in Baglietto et.al: loss 0.2045 using parameter settings from (b). (e) Examples of X_1 vs X_0 strategies using parameter settings from (c) that achieved a loss within range: 0.31 to 0.4658. (f) Examples of X_1 vs X_0 strategies using parameter settings from (c) that achieved a loss within range: 0.8 to 1.

Our approach: Lattice layer to bias networks towards good strategies

In an effort to make it easier to learn good control strategies we would like to bias the networks to make it more likely that they learn slopey-quantization-like strategies that perform well.

We achieve this by introducing a "lattice layer" at the beginning as illustrated in Figure 5.2. The lattice layer is parameterized by it width w, and takes an input X_0 which it splits it into two components, the bin-centre (X_0^c) and the offset in the bin (X_0^{off}), where

$$\begin{aligned} X_0^c &= \left\lfloor \frac{X_0}{w} \right\rfloor w + \frac{w}{2} \\ X_0^{off} &= X_0 - X_0^c. \end{aligned}$$

The idea is that the controller has binned the entire input space and will choose a control based on the bin and the position of the input within the bin. This makes it easier to learn slopey-quantization-like strategies as in subfigure(d) of Figure 5.1 since X_0^c can be used to determine which section we are in and X_0^{off} can be used to generate the slope for that section.

The outputs of the lattice layer are differentiable almost everywhere with respect to the width parameter w, and thus the width can be trained using gradient methods as well.

Throughout this training, we use an approximation to the conditional expectation as the second controller. Because for a slopey quantizer with non-zero slope the X_1 distribution isn't discrete, we cannot use an exact calculation for the conditional expectation. To get the approximate conditional expectation in a differentiable form we do the following. First, we note that the conditional expectation that we are interested in is:

$$\begin{aligned} \mathbb{E}[X_1 \mid Y_2 = y_2] &= \int_{x_1} x_1 p(x_1 \mid y_2) dx_1 \\ &= \int_{x_1} x_1 \frac{p(y_2 \mid x_1) p(x_1)}{p(y_2)} dx_1 \\ &= \int_{x_1} x_1 \frac{p(y_2 \mid x_1) p(x_1)}{\int_{x_1} p(y_2 \mid x_1) p(x_1) dx_1} dx_1, \end{aligned}$$

where we have used the Bayes rule to express $p(x_1 \mid y_2)$ in terms of $p(y_2 \mid x_1)$. Then we note that since $Y_2 = X_1 + Z$ and $Z \sim \mathcal{N}(0, 1)$ we have,

$$p(y_2 \mid x_1) = \frac{1}{\sqrt{2\pi}} e^{(y_2 - x_1)^2}.$$

Finally we approximate the integral $\int_{x_1} p(y_2 \mid x_1) p(x_1) dx_1$ by the mean of $p(y_2 \mid x_1)$ computed over a batch size of 1000 and similarly we also approximate the integral $\int_{x_1} x_1 p(y_2 \mid x_1) p(x_1) / p(y_2) dx_1$. The power of the automatic differentiation engine in PyTorch is that this gives rise to a second controller through which gradients could flow to the first controller.

We use the Adam optimizer [72] to train our neural network. More details about the training process are provided in Section III.C of [132].

Figure 5.2 visualizes our results. By looking at the histogram of losses for $k^2 = 0.04$ in subfigure (b) we see that now a majority of seeds achieve a low loss close to 0.2 as compared

to what we saw in subfigure (b) in Figure 5.1 and thus the lattice layer allows us to more robustly find a low loss strategy. Subfigure (c) plots the histogram of losses for $k^2 = 0.15$ and here too we notice that a majority of seeds achieve low loss below 0.5 when using the lattice layer. Subfigure (d) visualizes the strategy for the first controller that led to minimum loss of 0.18 when $k^2 = 0.04$. Subfigure (e) and (f) show other strategies that achieved a loss of 0.19 and 0.40 respectively for $k^2 = 0.04$.

Notice that the minimum loss of 0.18 achieved using our architecture with a lattice layer is very similar to the minimum loss of 0.20 obtained while using the architecture from [8] and this suggests that the reason the lattice layer helps us find a good strategy is not because it increases model capacity or allows us to express some strategies not possible without the lattice layer but it helps by biasing our network in such a manner that we can easily find strategies that enable slopey quantization.

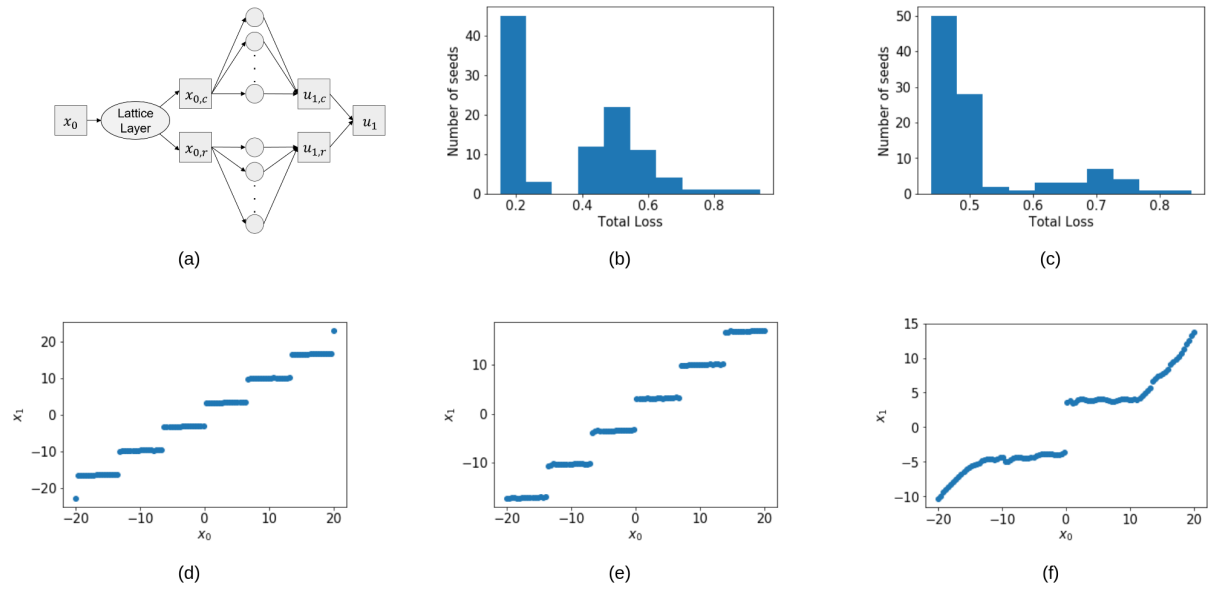


Figure 5.2. (a) Architecture for controller 1. Controller 2 uses approximate conditional expectation; $\mathbb{E}[X|Y]$. (b) Histogram of total losses of 100 seeds with $k^2 = 0.04$, $\sigma_X = 5$. (c) Histogram of total losses of 100 seeds with $k^2 = 0.15$, $\sigma_X = 5$. (d) Minimum loss strategy X_1 vs. X_0 that achieves loss = 0.18 using parameter settings in (b) . (e) Example X_1 vs. X_0 strategy that achieves total loss=0.19 using parameter settings from (b). (f) Example X_1 vs. X_0 strategy that achieves total loss=0.40 using parameter settings from (b).

Vector Witsenhausen problem

As pointed out in [55, 54], the Witsenhausen problem naturally extends to higher dimensions by making X_0 vector valued with m dimensions and an i.i.d. initial condition across those dimensions, and the cost functions normalized by $\frac{1}{m}$. These can be thought of as m indepen-

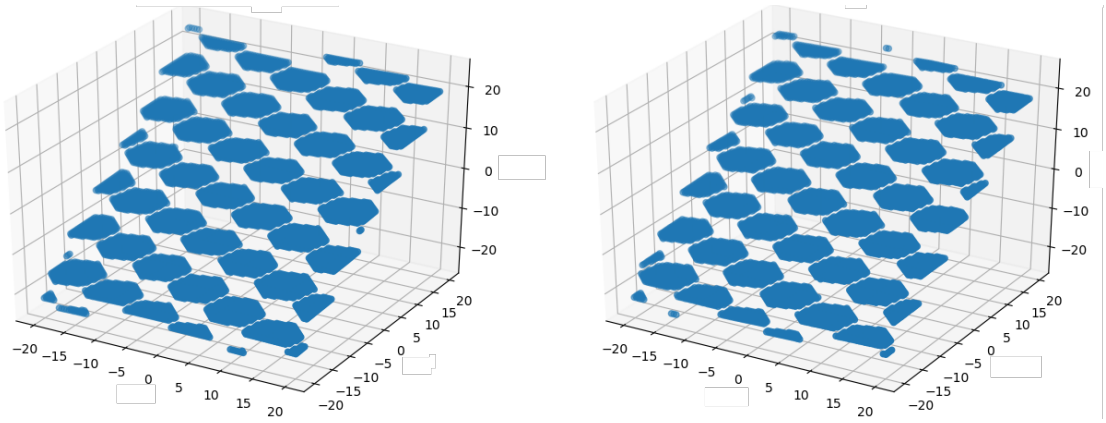


Figure 5.3. (Best) Final strategy for 2D Witsenhausen, with $k^2 = 0.04$, $\sigma_X = 5$. Learned using a regular hexagon lattice and the linear network architecture with 4 lattice layers at input. We used a learning rate=0.01, batch size=128, and iterations=3×10^4. Depicted is the best run out of 100 random initializations. XY-plane is the input to controller 1; Z-plane is one dimension of the X_1 state (left and right plots show the two different coordinates). Loss Stage 1 = 0.1280 ; Loss Stage 2 = 0.0248; Total Loss = 0.1527. Loss is estimated using 2×10^4 test data points.

dent parallel copies of the Witsenhausen counterexample but with the controllers allowed to peek at what is going on in the parallel instantiations before committing to an action for their particular copy. One of the main insights of [55, 54] is that just as vector quantization can do better than scalar quantization in pure lossy source coding problems even for i.i.d. sources, and just as error-correcting codes can do better with larger block-lengths even if they face independent noise in each channel, so also performance can in principle improve in vector versions of the Witsenhausen counterexample compared to the scalar version.

The relative success of this approach of introducing explicit lattice layers prompted us to see if this could be done for higher dimensions as well. As is discussed in [54], the best (in terms of packing/covering ratios) lattices in higher dimensions are not just replications of the natural 1D lattice provided by the integers. For example, in two-dimensional space, the natural lattice is that given by hexagons of side-length w. We implement a differentiable hexagonal lattice layer in PyTorch.[1] By using a slopey quantization strategy, the 2D loss can be reduced somewhat, as Figure 5.3 illustrates where in the 2-dimensional vector problem we achieved a minimum loss of 0.15 as compared to the minimum loss of 0.19 for the scalar problem.

[1]Given a regular hexagonal lattice of width w and a point X_0 the two-dimensional bin centers and and the distance of the point X_0 from the bin centers can be expressed in terms of w. Subsequently w is also another parameter that can be learned by the network. In our work in addition to the width w we also allow for these hexagon lattices to be rotated and learn the rotated angle as another parameter.

Discussion

Thus, we conclude that the implicit bias of the network architecture plays an important role not only in finding good strategies but also in how robustly/easily we can find such strategies. For the Witsenhausen problem, the implicit bias came from the lattice layer and was inspired by our knowledge about the problem domain and the insight that good strategies resembled slopey-quantization. Can a similar idea of selecting an appropriate implicit bias based on knowledge about problem domain be applied to learn control strategies for other control problems? The rest of the chapter explores this question by studying one particular control system, a linear system with multiplicative observation noise. While the resulting architectures and training procedure turn out to be more complex as compared to that of the Witsenhausen problem we will see that biasing our controller intelligently enables us to learn better control strategies.

5.3 Multiplicative noise system: Problem setup

We consider the discrete time system $\mathcal{S}_a$, with initial state $X_0 \sim \mathcal{N}(0,1)$, with system dynamics governed by:

$$X_{n+1} = aX_n - U_n \tag{5.1}$$
$$Y_n = Z_n X_n. \tag{5.2}$$

At timestep n, the system state is X_n. A controller observes this state over a multiplicative channel, i.e., $Y_n = Z_n X_n$. We think of Z_n as multiplicative noise. The Z_n's are drawn i.i.d. from a known distribution, however their realizations are unknown to the controller. We focus on $Z_n \sim \mathcal{N}(0,1)$ in this work, but the ideas generalize. The controller can determine the control at timestep n, U_n, as a function of the current and past observations, $Y_0, Y_1, \ldots, Y_n$, i.e. $U_n = \pi_n(Y_0, Y_1, \ldots, Y_n)$ where $\pi_n : \mathbb{R}^n \to \mathbb{R}$ is the control strategy at timestep n. We assume $a \in \mathbb{R}^+$ is fixed and known, and our goal is second-moment stability as defined below.

Definition 10. *The system $\mathcal{S}_a$ is stable in second-moment sense if* $\sup_n \mathbb{E}\left[|X_n|^2\right] < \infty$.

We are interested in the following questions: how can we learn control strategies π_n that enable us to stabilize the system? And, what is the largest a for which we can stabilize the system in a second-moment sense?

It is notable that the optimal linear strategy for this system is one that outputs $U_n = 0$ for all n, however non-linear strategies can significantly (and unboundedly) improve on the performance of the linear strategy [42]. Can we use neural networks to learn non-linear strategies like we did for the Witsenhausen problem (Section 5.2)?

5.4 Challenges while using neural networks to learn control strategies

Compared to the Witsenhausen problem we have some key additional challenges while learning a control strategy to stabilize the system in 5.1. First, our controller must be able to output the control at any timestep n however large. Thus, we cannot train a different neural network to output the control at each timestep. Second, practically, we can only train our control strategy for a finite horizon N. However, our control strategy must *generalize*, i.e. it should continue to decay the state of the system for timesteps beyond the training horizon.

To address these challenges, we utilize a periodic control structure and a greedy training procedure coupled with input-output scaling across time that enables us to learn control strategies that generalize well.

5.5 Our architecture and training procedure

Our control strategy is parameterized by memory, M, period P and greedy training horizon G. We say that a strategy uses memory M if it uses the values of observations $Y_n, Y_{n-1}, \ldots, Y_{n-M+1}$ to determine control action U_n. So a memory-1 controller can use only the current observation.

Next, we allow the system to periodically cycle through different neural networks as controllers. The parameter P denotes the number of distinct controllers we can use. The value of $n \pmod P$ is used to determine the controller that outputs control action at timestep n. Figure 5.4 provides an illustration of a memory-2, period-2 controller where we use a simple one hidden layer architecture for each network with 20 hidden units and ReLU activation.

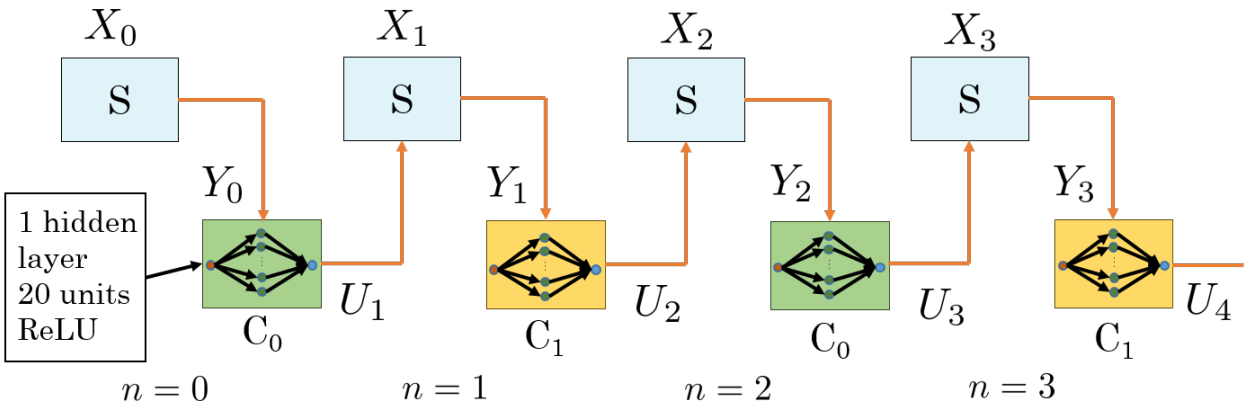


Figure 5.4. A memoryless ($M = 1$), 2-periodic controller. We alternate between the networks C_0 and C_1 for even and odd timesteps respectively.

We break the training into stages of length G and greedily minimize the second moment of the true state at the end of each stage. Figure 5.5 visualizes our greedy training procedure.

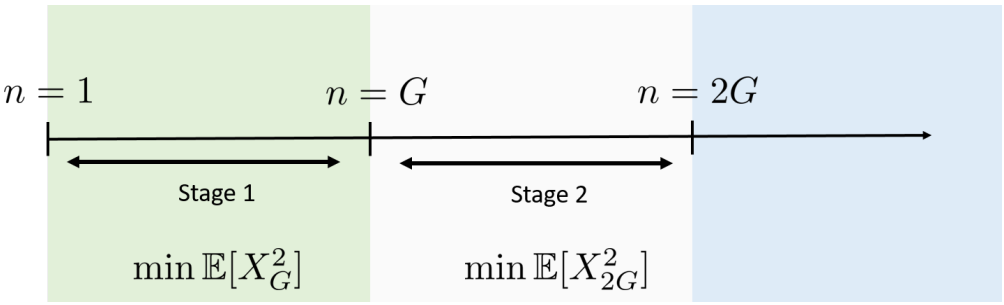


Figure 5.5: Greedy training procedure. We minimize the second moment every G steps.

More details about the control structure and training procedure are provided in Section 5.7. In the rest of the sections we use the M-P-G terminology to name our neural-network-based strategies; as an example M2-P2-G2 refers to a memory-2, period-2 controller that is trained greedily in stages of length 2.

Our goal is to try and close the gap between the achievability and the converse observed in [42] by determining what is the maximum growth factor a for which we can learn a control strategy that stabilizes the system $\mathcal{S}_a$ in second moment sense.

For simplicity, we focus on the case where $a = 1$, and consider the related system $\mathcal{S}$ given as:

$$X_{n+1} = X_n - U_n, \tag{5.3}$$
$$Y_n = Z_n X_n. \tag{5.4}$$

Our training loss function is engineered to minimize $\mathbb{E}[X_n^2]$ for this system. In Section 5.12 we show that how fast $\mathbb{E}[X_n^2]$ decays for system $\mathcal{S}$ is related to maximum growth factors a for which the system $\mathcal{S}_a$ can be stabilized. Faster rates of decay correspond to larger stabilizable growth factors.

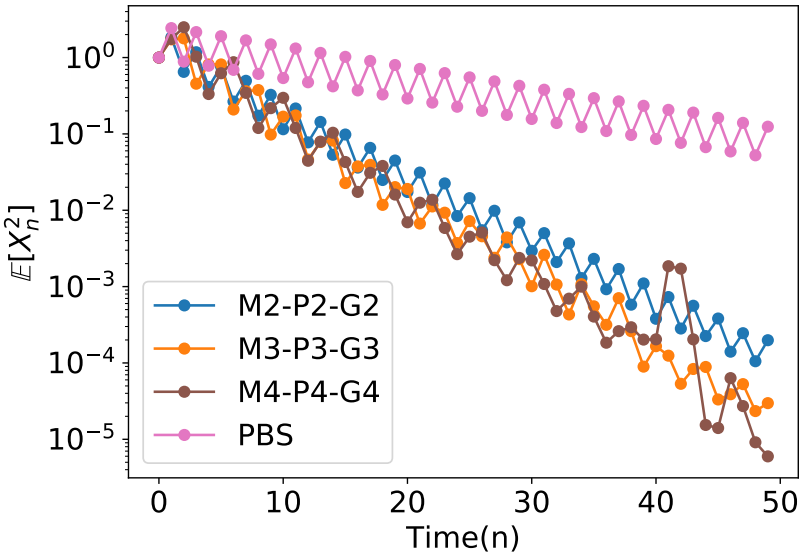
5.6 Main results

Next we present our two main results.

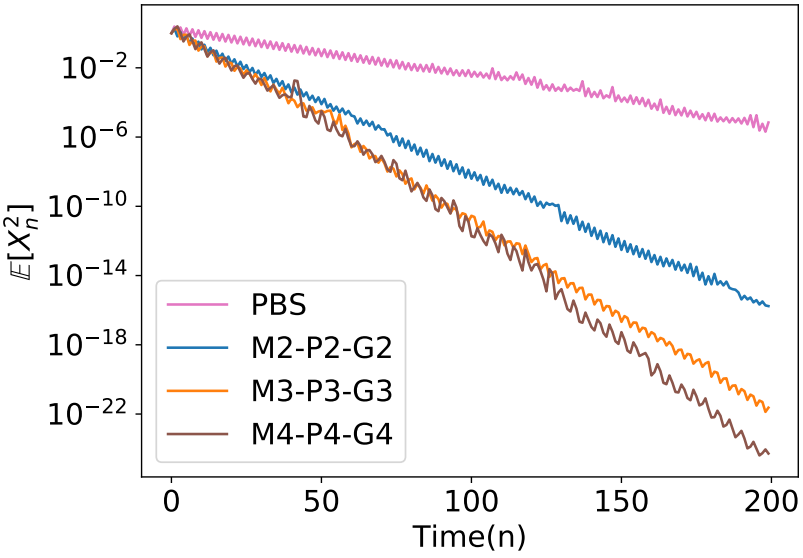
Neural-network strategies outperform hand-crafted strategies

Figure 5.6, which plots the second moment of the state vs the timestep, shows that neural-network-based strategies can outperform the hand-crafted previously best known strategy (PBS) [42]. The performance of the PBS plotted here comes from optimizing over the parameters of the strategy from [42].

Further, the neural-network-based control strategies are robust across the samples from the batch as can be seen from Figure 5.7 that plots the histogram of the absolute value of state for different timesteps. We see that for larger timesteps the absolute value of state is



(a) $n = 50$ timesteps



(b) $n = 200$ timesteps

Figure 5.6. Average second moment vs timestep for neural-network-based control strategies with memory 2, 3 and 4 and the previous best known strategy (PBS). The memory-2 strategy M2-P2-G2 outperforms the PBS. As we increase M we can achieve faster decay. The gap in the performance for memory-2 and memory-3 strategies is much larger than that between memory-3 and memory-4 strategies.

concentrated on smaller values. Further, the first moment of the state decreases with time as well as shown in Figure 5.8.

We tabulate the maximum growth factor that can be stabilized for different strategies in Table 5.1 by using the relationship between the decay rate of $\mathbb{E}[X_n^2]$ in system $\mathcal{S}$ to the maximum stabilizable growth rate for system $\mathcal{S}_a$ from Section 5.12. Restricting to memory-2 control strategies, M2-P2-G2 and can stabilize growth factors up to 1.097 while the PBS from [42] (which also uses memory-2) only stabilizes up to growth factors of 1.032. Increasing the memory of the control strategies further improves the performance of the neural-network-trained controllers and our best strategy can stabilize growth factors up to 1.156. However, there are diminishing returns to increasing memory and we elaborate more on this in Section 5.9.

Table 5.1: Maximum Growth Factors

Strategy	a^*
PBS	1.032
M1-P2-G4-FIT	1.025
M1-P2-G4	1.026
M1-P3-G6	1.026
M2-P2-G2-FIT	1.097
M2-P2-G2	1.097
M3-P2-G2	1.115
M3-P3-G3	1.137
M4-P4-G4	1.156

Neural-network strategies are well-structured and interpretable

Our choice of a periodic controller architecture and greedy training algorithm that enables us to learn well-structured interpretable control strategies.

We are able to understand the control strategies M1-P2-G4 and M2-P2-G4 as a linear combination of a few simple features as elaborate on in Section 5.8. For the memory-1 controller, M1-P2-G4 we do a piecewise linear fit while for the memory-2 controller, M2-P2-G4 we use a slightly more complicated set of features.

Further, our control strategies have a "probe" and then "minimize" structure, where for some timesteps we utilize a control that can increase the state magnitude, but at later timestep the observations are reduce to decrease the state magnitude.. The system probing suggests there is an element of "active" learning our learned control strategies.

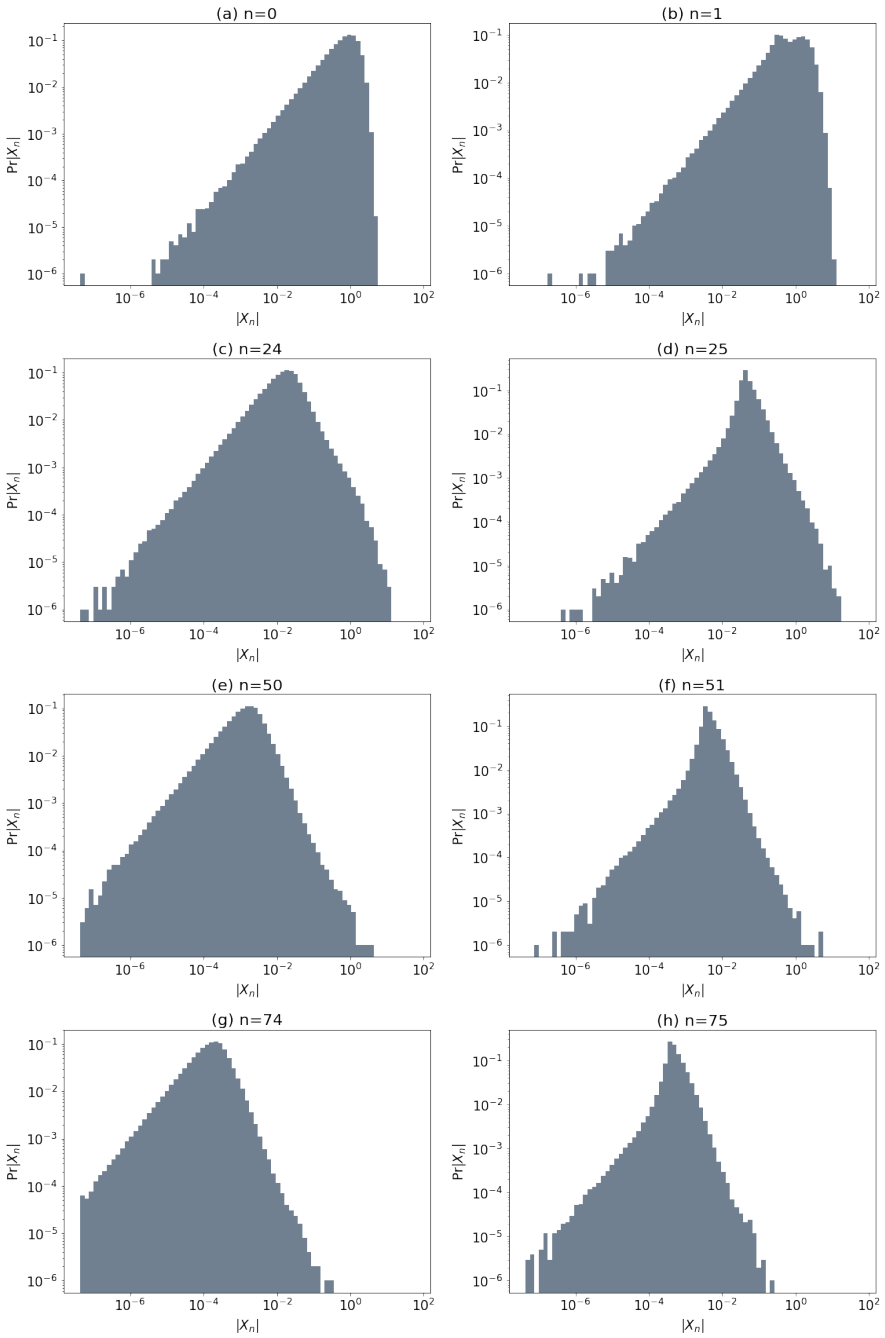


Figure 5.7. Histogram of the absolute value of state for various timesteps when using the $M2 - P2 - G2$ control strategy. Notice the shape of the histogram is different for even and odd timesteps since we use a different controller for odd and even timesteps. For larger timesteps the absolute value is concentrated on smaller values since our control strategy leads to decaying state.

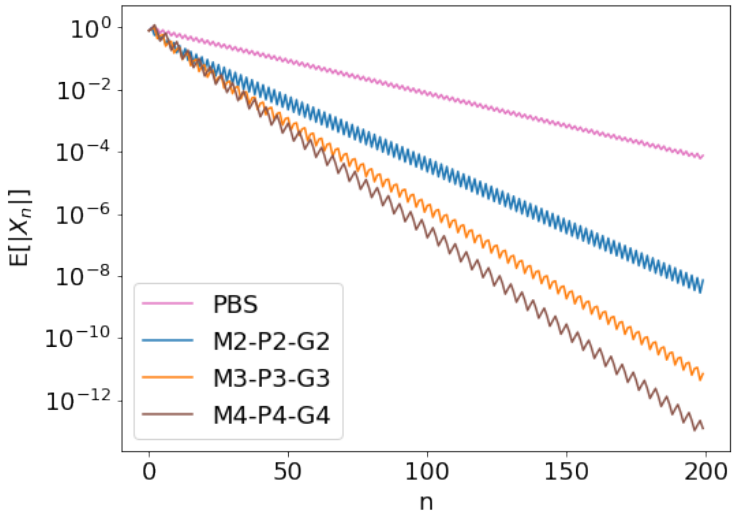


Figure 5.8. First moment of state vs timestep for strategies with different M, P, G values. Comparing this plot to Figure 5.6 we see that strategies that perform better in minimizing second moment also perform better in minimizing first moment.

5.7 Methods

In this section we describe our control structure and training procedure in detail.

Control structure

Each control strategy consists of a set of P networks. At timestep n the network corresponding to $n \pmod P$ is used to generate the control U_n based on the past M memory of observations $Y_n, \dots Y_{n-M+1}$. The set of networks over the period P form the control strategy that aims to minimize the second moment of the system state N timesteps in the future. We use a simple one hidden layer architecture for each network with 20 hidden units and ReLU activation as illustrated in Figure 5.4.

Since we periodically reuse the same network for control, it is important to consider the inputs and outputs to the network carefully. A good control strategy will decrease the magnitude of the state X_n and thus also the magnitude of the observations Y_n and the required U_n. While an observation value of 0.5 might be typical for timestep 0, it would be a very atypical observation at timestep 100, and hence must be treated differently by the controller. To deal with this we scale inputs and outputs of the networks as described below.

We use an exponential scaling factor s_n given as:

$$s_n = \alpha^{\lfloor \frac{n}{P} \rfloor}.$$

Note that we use the floor, $\lfloor n/P \rfloor$ because our control minimizes the second moment of state only every P steps as can be seen in Figure 5.6.

At timestep n, we scale the observations Y_n by s_n to give $\widetilde{Y}_n$ as:

$$\widetilde{Y}_i = \frac{Y_i}{s_n} \quad i = n - M + 1, \ldots, n.$$

Note that at timestep n, we scale both the current and previous observations by the scaling factor corresponding to timestep n in order to preserve the relative order between these observations. Similarly we scale down the network's output $\widetilde{U}_n$ to U_n before applying the control action to the system with $U_n = s_n \widetilde{U}_n$. Figure 5.9 illustrates the input-output scaling.

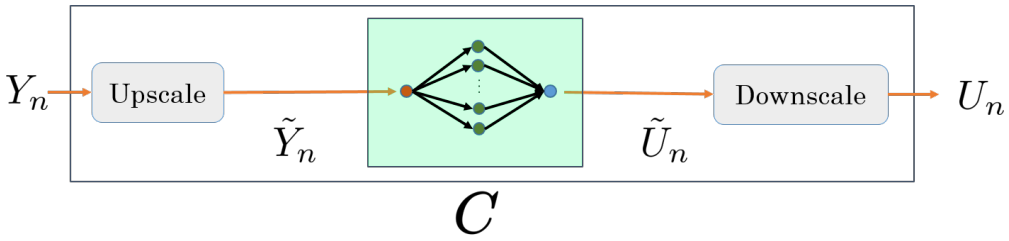


Figure 5.9. Scaling to preserve scale of inputs and outputs across time. The inputs to the network (i.e the observations Y_n) are up-scaled while the outputs of the network $\widetilde{U}_n$ are down-scaled.

We identify the best α by a hyperparameter search and provide the values used in Section 5.10. A value of α that is too high or too low leads to poor control strategies that do not generalize. The optimal α is one that leads to approximately constant second moment of the inputs to the neural networks after the rescaling.

Training procedure

To train the neural networks we break the training horizon N into stages of length G and greedily minimize the second moment of the true state every G steps as illustrated in Figure 5.5. We use truncated backpropagation through time and prevent the flow of gradients across stages [149].

We choose $G = kP$ for some positive integer k. As an example for $M = 2, P = 2, G = 2$ the first stage involves minimizing $\mathbb{E}[X_2^2]$ and the second stage involves minimizing $\mathbb{E}[X_4^2]$. During the second stage we treat Y_2 and Y_1 as fixed constants not dependent on the parameters of the neural networks.

Our control structure and training procedure resembles that of a stateless recurrent neural network [128] with our scaling procedure and periodic control structure performing the role of the state. The structure that we impose makes it easier to train our control strategies as compared to training recurrent neural networks. Further, this approach allows us to learn control strategies that generalize well by continuing to decrease the second moment for timesteps beyond the training horizon as shown in Figure 5.10.

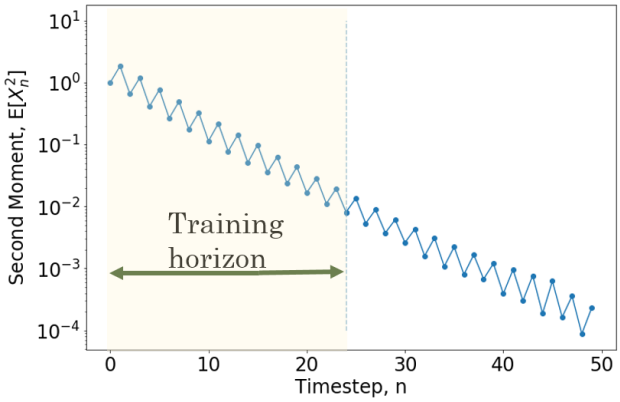


Figure 5.10. The learned control strategy M2-P2-G2 generalizes and continues to decrease the second moment for timesteps beyond the training horizon of 24.

We use a training batch size of 4000 and run rollouts of the system (5.3) for a total training horizon of 24. We train our neural networks for 10000 iterations using the Adam optimizer with a learning rate of 10^{-4}.

Robustness

To test our controllers we run rollouts of the strategies for batches of 10^6. Note that though this batch size is large, it is not large enough to see certain types of inputs. This might make some strategies appear to be successful even though they should fail. For instance, the probability that all noise realizations are positive for a rollout up to timestep 50 is roughly 9×10^{-16} and we would require a batch size of around 10^{17} to consistently see such inputs.

However, we believe this is not an issue with our strategies since the rate of decrease of the second moment is consistent across time since our controller does a minimization of second moment every P steps. Our batch size is large enough that the empirical test performance is an accurate indicator for the true test performance for each period length. Figure 5.11 shows the variation in the second moment of the state for the batch size of 10^6 compared to a batch size of 10^4. We see that for the batch size of 10^6 the second moment metric is quite robust and there is not much deviation across trials for small values of n up to 40.

5.8 Neural-network-based strategies

This section explores two successful control strategies in detail. We see that these strategies can be understood as linear combination of a few simple features.

M1-P2-G4

This control strategy uses only the current observation to compute the control and has a 2-periodic structure. Since $G = 4$ we minimize the second moment in stages of length 4.

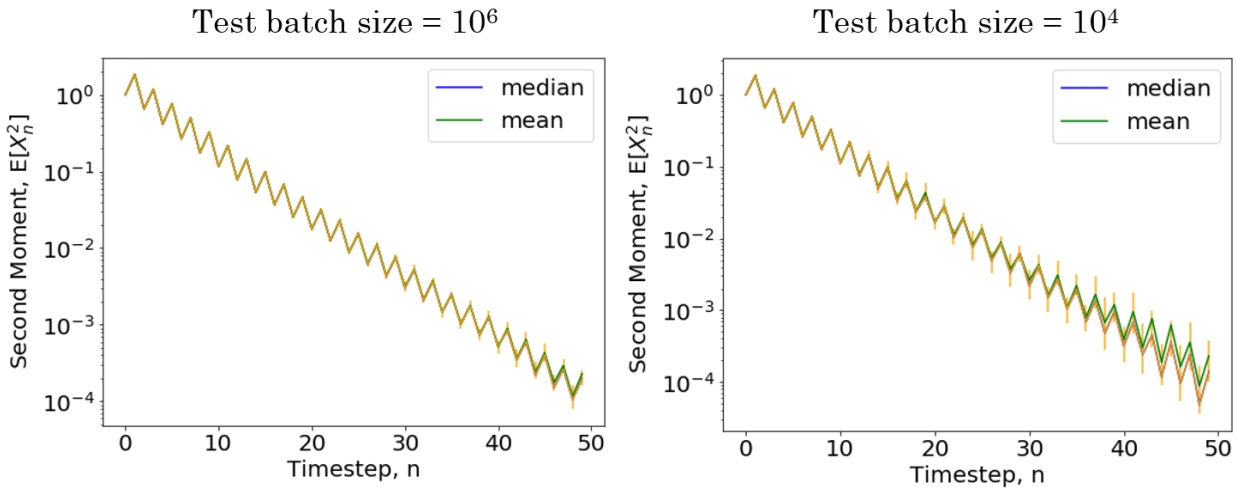


Figure 5.11. Variation of second moment over 20 different test batches while using the M1-P2-G4 controller. The orange vertical lines denote the error bars corresponding to the 10^{th} and 90^{th} percentiles. For a test batch size of 10^6, the second moment metric is robust upto $n = 40$ and the error bars are small. Using a smaller batch size of 10^4 however leads to larger error bars and more variation across test batches.

It uses two networks, one for even timesteps and one for odd timesteps, as described in Sec. 5.7. Since the magnitude of states, observations, and controls for the system decay with time, we need to scale them appropriately as described in detail in Sec. 5.7. The observations, Y_n, are scaled up to $\widetilde{Y_n}/s_n$ before being fed into the network. The output of the network, $\widetilde{U}_n$, is scaled down to $U_n = s_n\widetilde{U}_n$ before being applied to the system. The red curves in Figure 5.12 show $\widetilde{U}_n$, the output of the neural network, as a function of $\widetilde{Y}_n$, the input of the neural network, at even and odd timesteps.

A first observation about this strategy is that it mostly ignores the sign of $\widetilde{Y}_n$. This makes sense, since the zero-mean multiplicative noise means there is no information in the sign of $\widetilde{Y}_n$. Further, it flips sign at even and odd times and seems to have a "probe" and then "minimize" structure, where at the even timestep it sends a test control that might increase the state magnitude, but uses the observation from this to reduce the magnitude at the following timestep. The system probing suggests there is an element of "active" learning in the strategy.

Based on the shape of the plot, we choose to use a piecewise linear fit (M1-P2-G4-FIT) for the function, and the best fit (using input range $[-5, 5]$) is shown via the green curves in Figure 5.12. We use the following features for the fit: $\max(-\widetilde{Y}_n + h_L, 0)$, $\max(-\widetilde{Y}_n, 0)$, $\max(\widetilde{Y}_n, 0)$, $\max(\widetilde{Y}_n - h_R, 0)$, and 1 (i.e. a bias term). The parameters h_R and h_L, where our piecewise linear fit changes slope, as well as the weights were identified using non-linear least-squares for both the even and odd time control strategies and are listed in Section 5.11.

We test the performance of M1-P2-G4-FIT on the actual system and find that it has performance very close to that of the neural network based strategy (see Figure 5.13). We

see that both strategies can stabilize similar growth factors, 1.025 vs 1.026 (Table 5.1).

For both the neural-network-based strategy and the fit strategy the effective function that relates Y_n to U_n changes with timestep n due to our scaling operation. To check that the scaled inputs $\widetilde{Y}_n$, continue to lie in the range that was used to fit the piecewise linear strategy for different n, we plot the 95$^{\text{th}}$ percentile values of $|\widetilde{Y}_n|$ in Figure 5.14. We see that in around 30 timesteps the 95$^{\text{th}}$ percentile values stabilize to a consistent range depicted by the shaded region in Figure 5.12 and the outer regions are used only for timesteps $n < 30$ and for outliers.

M2-P2-G2

Next we consider a memory-2 control strategy with period 2. The successfully trained neural networks use the two most recent scaled observations, $\widetilde{Y}_n$ and $\widetilde{Y}_{n-1}$, to output $\widetilde{U_n}$, and plots for these functions are shown in Figure 5.15.

If we fix $\widetilde{Y}_{n-1}$ and look at the functional relationship between $\widetilde{Y}_n$ and $\widetilde{U}_n$, we see similarities to the memory-1 strategy. $\widetilde{U}_n$ is largely indifferent to the sign of $\widetilde{Y}_n$ and increases in the magnitude based on the magnitude of $\widetilde{Y}_n$. However, the relationship between $\widetilde{Y}_n$ and $\widetilde{Y}_{n-1}$ plays an important role in the control strategy now. If we look at the plane spanned by $\widetilde{Y}_n$ and $\widetilde{Y}_{n-1}$ then we can identify four lines in this plane (angles) where the behaviour of the strategy changes. These lines correspond to the creases on the $\widetilde{U}_n$ surface. Motivated by this observation we fit the function (M2-P2-G2-FIT) using the following features: $|\widetilde{Y}_{n-1}|$

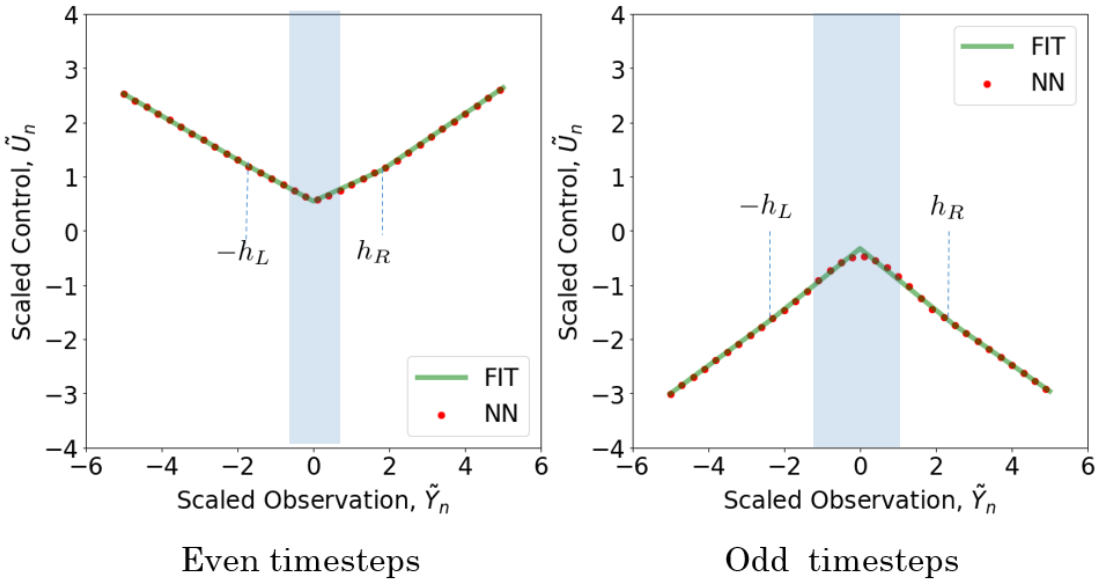


Figure 5.12. Visualizing the memory-1 neural network strategy, $\widetilde{U}_n = f(\widetilde{Y}_n)$, for even and odd timesteps. The fit strategy shown in green closely resembles the neural network strategy shown in red. For $n > 30$, 95% of $\widetilde{Y}$'s fed to the neural network lie in the shaded region.

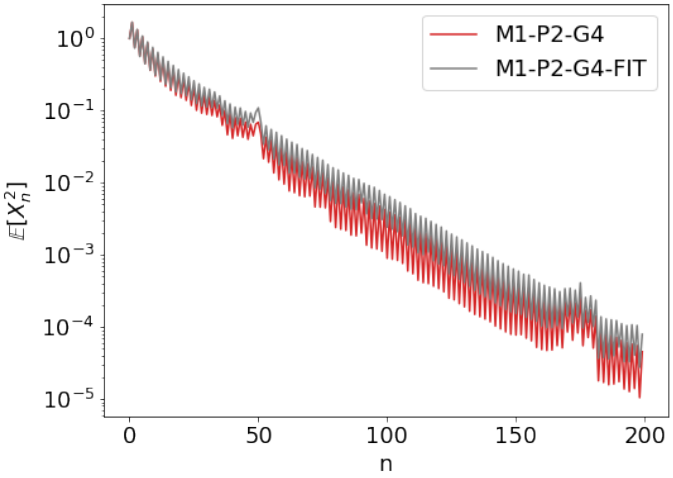


Figure 5.13. Performance comparison of the fit strategy, M1-P2-G4-FIT to neural network strategy M1-P2-G4. Both strategies exhibit similar performance.

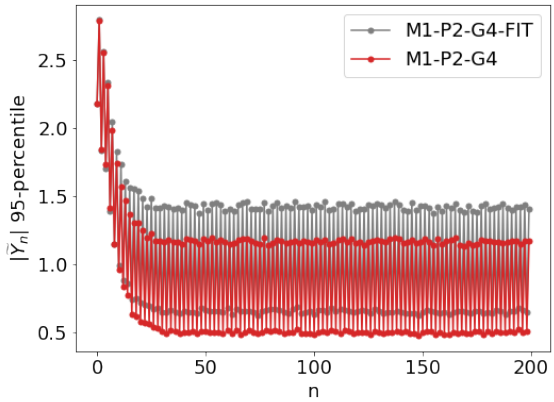


Figure 5.14. 95^{th} percentile value of $|\widetilde{Y}_n|$ (the dots) fed to the neural network with timestep : There is a difference in the even and odd timesteps as expected for $P = 2$. What is notable is that after about 30 timesteps the inputs stabilize to being in close ranges for both the neural-network-trained controller and the piece-wise linear fit controller. This suggests that good control strategies eventually stabilize the input distribution to the network (up to scaling).

and $|\widetilde{Y}_n|$, as well as, $|\cos(\theta_1)\widetilde{Y}_{n-1} - \sin(\theta_1)\widetilde{Y}_n|$, $|\cos(\theta_2)\widetilde{Y}_{n-1} + \sin(\theta_2)\widetilde{Y}_n|$ (i.e. oriented along diagonal lines $Y_n = \pm\cot(\theta)Y_{n-1}$) and 1 (for bias). We still lack a clean explanation for the exact reason why this strategy works.

The strategy generated by the fit exhibits similar performance to the network as shown in Figure 5.16, and can stabilize similar growth factors as provided in Table 5.1. Like the $M = 1$ strategy, this $M = 2$ strategy exhibits the same zigzag behavior at even and odd timesteps, and the range of scaled inputs also stabilizes after about $n = 30$ timesteps.

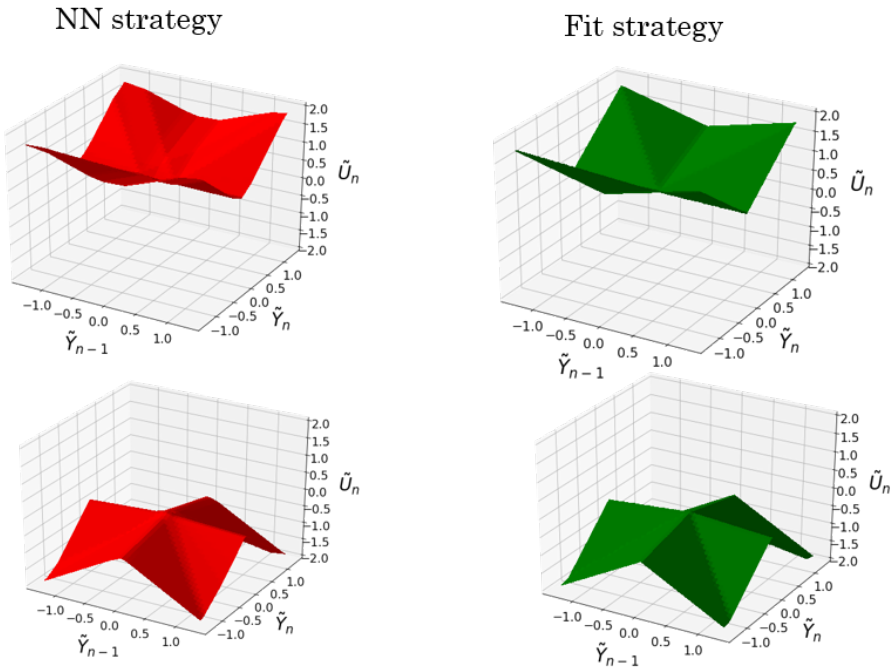


Figure 5.15. Visualizing the memory-2 neural network strategy alongside the fit strategy. This figure shows the value of $\widetilde{U}_n$ at even (top row) and odd (bottom row) timesteps as a function of $\widetilde{Y}_n$ and $\widetilde{Y}_{n-1}$. We observe that the fit strategy visually resembles the neural network strategy.

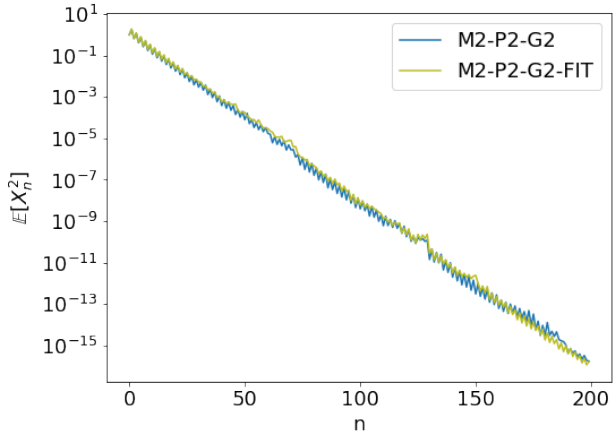
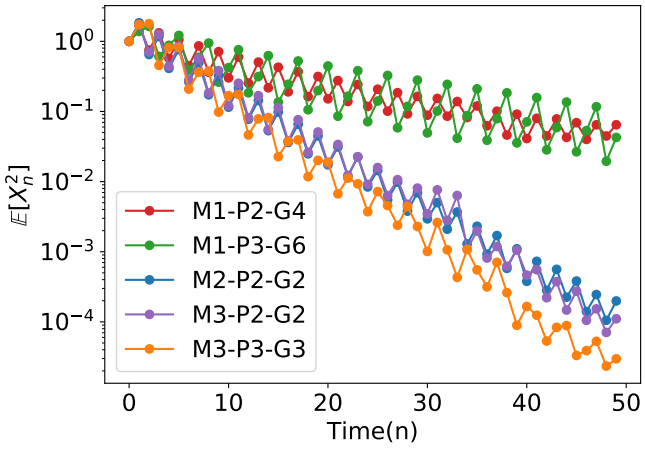


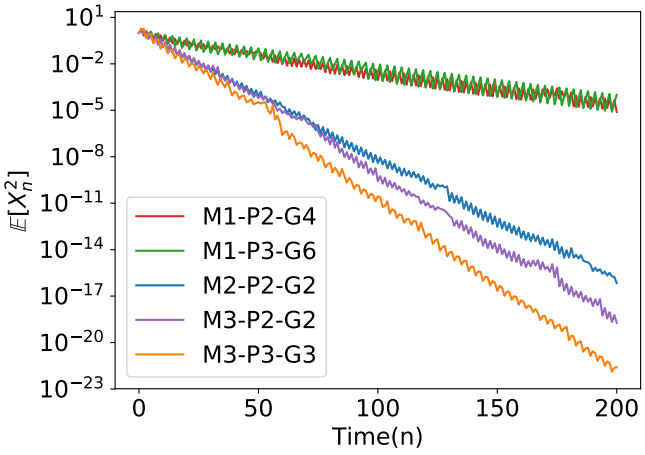
Figure 5.16. Performance comparison of the fit strategy to the memory-2 neural network generated strategy. The performance of these strategies and the maximum growth factor that these can stabilize are similar.

5.9 Effects of the parameters M, P and G

From Figure 5.6 in Section 5.6 it is clear that increasing memory (M), controller period (P) and planning horizon (G) improve performance. This section investigates the effects of these parameters in more detail and Figure 5.17 summarizes the results.



(a) Short rollout up to $n = 50$



(b) Long rollout up to $n = 200$

Figure 5.17. Comparison of strategies with different M, P, G values. For memory-1 control strategies $P = 2$ and $P = 3$ result in similar performance but how the strategy alternates between probing and minimization step varies. Increasing M while keeping P, G constant results in better performance but further improvement can be obtained by increasing P, G as well.

We observe that both of the memory-1 strategies, M1-P2-G4 and M1-P3-G6 have similar performance; and increasing the period P or the horizon G while keeping M constant does

not seem to improve performance. We notice that when G is a multiple of P, the value of P plays a role in the structure of the strategy. The last action during the period (i.e. when $n = -1 \pmod{P}$) is always the minimizing control action, whereas earlier actions can be thought of as probing actions.

For memory-1 strategies we observed that setting $G = P$ results in overfitting while training leading to poor test performance. Keeping M and P constant and increasing the value of G does not lead to better performance during training but alleviates the problem of overfitting by taking account of the fact that the same networks are being used across multiple periods, while minimizing the cost function. Thus the strategy M1-P2-G4 performed much better that M1-P2-G2 during testing.

We were unable to train the memory-1 control strategy to work with $P = 1$. Comparing all the strategies M2-P2-G2, M3-P2-G2, M3-P3-G3, we see that the control structure with the most information and degrees of freedom gave the best performance.

5.10 Values for scaling hyperparameter α

We list the alpha values for different strategies in Table. 5.2. We see that strategies that perform better and lead to faster decay of rates correspond to smaller values α. For the fit strategies we use the same values as the original neural-network-based strategy.

Table 5.2: α values

Strategy	α
M1-P2-G4-FIT	0.955
M1-P2-G4	0.955
M1-P3-G6	0.933
M2-P2-G2-FIT	0.832
M2-P2-G2	0.832
M3-P2-G2	0.808
M3-P3-G3	0.685
M4-P4-G4	0.551

5.11 Fit strategies

Fit strategy for M1-P2-G4

We can express the function that relates the scaled inputs $\widetilde{Y}_n$ to the network to the output of the network $\widetilde{U}_n$ as,

$$\begin{aligned}\mathrm{M1-P2-G4-FIT} = f_{fit}(\tilde{Y}) = (m_1 - m_2)\max(-\tilde{Y} + h_L, 0) + m_2\max(-\tilde{Y}, 0) \\ + m_3\max(\tilde{Y}, 0) + (m_4 - m_3)\max(\tilde{Y} - h_R) + b.\end{aligned}$$

The resultant values for the parameters $h_L, h_R, m_1, m_2, m_3, m_4$, and b for even and odd timestep networks are provided in Table 5.3.

Fit Parameter	Even time	Odd time
h_L	1.807	2.367
h_R	1.808	2.429
m_1	0.406	-0.512
m_2	0.378	-0.561
m_3	0.317	-0.568
m_4	0.476	-0.486
b	0.542	-0.329

Table 5.3: Parameters for M1-P2-G4 fit strategy

Fit strategy for M2-P2-G2

Since this is a memory-2 strategy, we can express the function that relates the scaled inputs $\widetilde{Y}_n, \widetilde{Y}_{n-1}$ to the output $\widetilde{U}_n$ as,

$$\begin{aligned}\mathrm{M2-P2-G2-FIT} = g_{fit}\left(\widetilde{Y}_n, \widetilde{Y}_{n-1}\right) = k_1\left|\widetilde{Y}_{n-1}\right| + k_2\left|\widetilde{Y}_n\right| + k_3\left|\cos(\theta_1)\widetilde{Y}_{n-1} - \sin(\theta_1)\widetilde{Y}_n\right| \\ + k_4\left|\cos(\theta_2)\widetilde{Y}_{n-1} + \sin(\theta_2)\widetilde{Y}_n\right| + b.\end{aligned}$$

The resultant values for the parameters $k_1, k_2, k_3, k_4, \theta_1, \theta_2$, and b for even and odd timestep networks are provided in Table 5.4.

In the next section, we connect the minimum decay factor of system $\mathcal{S}$ to maximum stabilizable growth factor for system $\mathcal{S}_a$.

Fit Parameter	Even time	Odd time
k_1	-0.455	0.863
k_2	-0.924	1.302
k_3	0.271	-0.548
k_4	0.279	-0.585
θ_1	0.654	0.856
θ_2	0.682	0.870
b	-0.675	0.236

Table 5.4: Parameters for M2-P2-G2 fit strategy

5.12 Connecting minimum decay factor of system $\mathcal{S}$ to maximum stabilizable growth factor for system $\mathcal{S}_a$

Recall the dynamics of system $\mathcal{S}$ and $\mathcal{S}_a$. For system $\mathcal{S}$, we have:

$$\begin{aligned} X_{n+1} &= X_n - U_n, \\ Y_n &= Z_n X_n. \end{aligned}$$

For system $\mathcal{S}_a$, we have:

$$\begin{aligned} X_{n+1} &= aX_n - U_n \\ Y_n &= Z_n X_n. \end{aligned}$$

In both cases $X_0 \sim \mathcal{N}(0,1)$ and $Z_n \sim \mathcal{N}(0,1)$ are i.i.d.

For system $\mathcal{S}$, we define the minimum decay factor as below.

Definition 11. *The minimum decay factor of $\mathcal{S}$ is given as:*

$$d^* = \limsup_{n\to\infty} \inf_{\pi_0,\pi_1,\ldots,\pi_n} \left(\frac{\mathbb{E}[|X_n|^2]}{\mathbb{E}[|X_0|^2]} \right)^{\frac{1}{2n}}.$$

The next theorem shows the relationship between the minimum decay factor for system $\mathcal{S}$ and the maximum growth factor a^* for which the system $\mathcal{S}_a$ can be stabilized.

Theorem 7. *Let d^* denote the minimum decay factor for the system $\mathcal{S}$. The system $\mathcal{S}_a$ can be stabilized for all $a < a^*$ and cannot be stabilized for any $a > a^*$ where $a^* = 1/d^*$.*

Proof. For the sake of the proof, let us label the states, observations, multiplicative noise and control actions for the system $\mathcal{S}_a$ as X_n^a, Y_n^a, Z_n^a and U_n^a respectively. We will use the notation X_n, Y_n, Z_n and U_n for the system $\mathcal{S}$. Consider a coupling of the two systems such that $X_0 = X_0^a$ and $Z_n^a = Z_n$ for all n.

Suppose the control action at timestep n, U_n, for system $\mathcal{S}$ is given by $U_n = \pi_n(Y_0, Y_1, \ldots, Y_n)$. Note that here $\pi_n : \mathbb{R}^n \to \mathbb{R}$ is a deterministic function given the realizations of the observations. Then, we construct the following control actions for system $\mathcal{S}_a$,

$$U_n^a = \pi_n^a(Y_0^a, Y_1^a, \ldots, Y_n^a), \tag{5.5}$$

where,

$$\pi_n^a(Y_0^a, Y_1^a, \ldots, Y_n^a) := a^{n+1}\pi_n\left(Y_0^a, \frac{Y_1^a}{a}, \ldots, \frac{Y_n^a}{a^n}\right). \tag{5.6}$$

We will prove by induction that under these sets of controls, the true state and observations for the two systems are related as,

$$X_n^a = a^n X_n, \quad Y_n^a = a^n Y_n.$$

Note that by our assumption on the initial state and noise realizations the base case for $n = 0$ is true. Now assume that the claim is true for $n \leq k$. We have,

$$\begin{aligned}
X_{k+1}^a &= aX_k^a - U_k^a \\
&= a^{k+1}X_k - a^{k+1}\pi_k\left(Y_0^a, \frac{Y_1^a}{a}, \ldots, \frac{Y_k^a}{a^k}\right) \\
&= a^{k+1}X_k - a^{k+1}\pi_k\left(Y_0, Y_1, \ldots, Y_k\right) \\
&= a^{k+1}(X_k - U_k) \\
&= a^{k+1}X_{k+1}.
\end{aligned}$$

Further since the noise realizations are same we have

$$Y_{k+1}^a = Z_{k+1}^a X_{k+1}^a = Z_{k+1}a^{k+1}X_{k+1} = a^{k+1}Y_{k+1}.$$

Thus the claim is true for $n = k + 1$ and this completes the inductive proof.

Next we will show that if the minimum decay factor for system $\mathcal{S}_a$ is d^* then system $\mathcal{S}_a$ can be stabilized for $a < 1/d^*$.

Suppose π_k^* minimizes $\left(\frac{\mathbb{E}[|X_k|^2]}{\mathbb{E}[|X_0|^2]}\right)^{\frac{1}{2n}}$ for each $0 \leq k \leq n$. Let π_n^* denote the minimizing control action for system $\mathcal{S}$ at timestep n and consider $(\pi_n^a)^*$ as the control action for system $\mathcal{S}_a$, where $(\pi_n^a)^*$ and π_n^* are related as in Equation (5.6).

From Definition 11 we have,

$$d^* = \limsup_{n\to\infty}\left(\frac{\mathbb{E}[|X_n|^2]}{\mathbb{E}[|X_0|^2]}\right)^{\frac{1}{2n}}.$$

Then by definition of lim sup we have for every $\epsilon > 0$, there exists N such that for all $n \geq N$,

$$d^* \geq \left(\frac{\mathbb{E}[|X_n|^2]}{\mathbb{E}[|X_0|^2]}\right)^{\frac{1}{2n}} - \epsilon.$$

Thus,

$$(d^* + \epsilon)^{2n} \geq \left(\frac{\mathbb{E}[|X_n|^2]}{\mathbb{E}[|X_0|^2]}\right) = \frac{1}{a^{2n}}\mathbb{E}[|X_n^a|^2],$$

since $\mathbb{E}[|X_0|^2] = 1$. Any $0 < a < 1/d^*$ can be written as $a = (1-\delta)(1/d^*)$ for some $0 < \delta < 1$. Thus we have,

$$\mathbb{E}[|X_n^a|^2] \leq a^{2n}(d^* + \epsilon)^{2n} = \left(1 - \delta + \frac{\epsilon(1-\delta)}{d^*}\right)^{2n}.$$

Since this bound holds for any $\epsilon > 0$, taking $\epsilon = \frac{\delta d^*}{2(1-\delta)} > 0$, there exists N such that for all $n \geq N$,

$$\mathbb{E}[|X_n^a|^2] \leq \left(1 - \frac{\delta}{2}\right)^{2n} < 1.$$

Further, because $\sup_{n<N} \mathbb{E}[|X_n^a|^2]$ is finite, we conclude that system $\mathcal{S}_a$ is stabilizable.

Next we will show that the system $\mathcal{S}_a$ cannot be stabilized for any $a > \frac{1}{d^*}$. Consider such an $a = \frac{1}{d^*}(1+\gamma)$ for some $\gamma > 0$. Suppose for contradiction there exists a set of control actions $\widetilde{\pi_0^a}, \widetilde{\pi_1^a}, \ldots, \widetilde{\pi_n^a}$ such that $\sup_n \mathbb{E}[|X_n^a|^2]$ is finite. Thus there exists $K < \infty$ such that for all n,

$$\mathbb{E}[|X_n^a|^2] \leq K.$$

Further using the set of control actions $\widetilde{\pi}_0, \widetilde{\pi}_1, \ldots, \widetilde{\pi_n}$ where $\widetilde{\pi_n^a}$ and $\widetilde{\pi_n}$ are related as in Equation (5.6) we have,

$$\begin{aligned}
&\mathbb{E}[|X_n^a|^2] \leq K \\
\implies\ & a^{2n}\mathbb{E}[|X_n|^2] \leq K \\
\implies\ & a^{2n}\frac{\mathbb{E}[|X_n|^2]}{\mathbb{E}[|X_0|^2]} \leq K \\
\implies\ & \left(\frac{\mathbb{E}[|X_n|^2]}{\mathbb{E}[|X_0|^2]}\right)^{\frac{1}{2n}} \leq (K)^{\frac{1}{2n}} a^{-1} = (K)^{\frac{1}{2n}} \frac{d^*}{1+\gamma}. \qquad (5.7)
\end{aligned}$$

Choose N such that for all $n \geq N$,

$$(K)^{\frac{1}{2n}} < (1+\gamma).$$

Note that such an N exists because K is finite and $\lim_{n\to\infty}(K)^{\frac{1}{2n}} = 1$. Since the upper bound in Equation (5.7) holds for all n, we have for $n \geq N$,

$$\left(\frac{\mathbb{E}[|X_n|^2]}{\mathbb{E}[|X_0|^2]}\right)^{\frac{1}{2n}} < d^*,$$

Thus, we have a set of control actions $\widetilde{\pi_0}, \widetilde{\pi_1}, \ldots, \widetilde{\pi_n}$ such that,

$$\limsup_{n\to\infty} \left(\frac{\mathbb{E}[|X_n|^2]}{\mathbb{E}[|X_0|^2]}\right)^{\frac{1}{2n}} < d^*$$

which is a contradiction since we assumed that d^* was the minimum decay factor. This completes the proof. □

5.13 Discussion

We find that by choosing the appropriate control structure and training procedure we can learn neural-network-based control strategies that can stabilize a multiplicative observation noise system and outperform hand-crafted strategies. Allowing strategies to use more memory improves performance but has diminishing returns. There is a structure and planning aspect to the learned strategies that can be expressed in terms of simple features. However there are many open questions. Can we exactly quantify how the memory and period of the controllers affects their performance and is there an optimal memory and period to use? How much better are the periodic controller architectures over other alternatives like recurrent neural networks or simple a feed-forward network that takes the current timestep as an input?

Can we better understand the probe and minimize aspect of the controllers? Can this provide insights into the fundamental communication bottlenecks imposed by multiplicative noise in control systems, for example, is there some amount of "active" learning that must be done in these systems by probing?

Chapter 6

Discussion and future directions

In this thesis, we discovered the importance of implicit bias of the architecture and learning algorithm for learning models that generalize well. It is not sufficient to consider whether we *can* learn a model that generalizes well (i.e. whether the model is rich enough to express the true underlying reality) but it is important to study whether we will *will* learn a model that generalizes well (i.e whether the training algorithm and architecture work in a manner that steer us towards a good solution).

In Chapters 2,3 and 4 we theoretically studied generalization for overparameterized linear models via a signal-processing-inspired perspective. Good generalization requires the true signal to be contained in or at least well approximated by a few directions that are *favored* while we perform the reconstruction. At the same time, there must be sufficiently many unimportant directions that can dissipate the training noise harmlessly. In Chapter 3, we proved the existence of a new asymptotic regime where good generalization is possible for the binary classification task but is not possible for the regression task. Similarly, in Chapter 4, we saw that multiclass classification too can succeed when regression fails as long as there are not too many classes.

In Chapter 5, we empirically investigated the use of non-linear neural network models to tackle two problems in control. First, for the 2-step Witsenhausen problem we observed that the use of a lattice layer provides the right implicit bias towards slopey quantization strategies that perform well. Second, for the infinite horizon problem of stabilizing the control system with multiplicative noise a periodic controller structure and a greedy training procedure along with input-output scaling enables us to learn control strategies that generalize well beyond the finite training horizon. An interesting question is whether we can understand how exactly the particular architectures and training procedures we used here shape the loss landscape and consequently make it easier to find a good solution (control strategy). Here, the challenge is in visualizing these high-dimensional loss functions. Understanding how the loss landscape is shaped in these control problems and comparing it to how the weighting of features in overparameterized linear model shapes the loss landscape can be an avenue of research that helps bridge from linear models (where we have a firm theoretical understanding) to non-linear neural network models where little is understood theoretically.

In our work, we observed the phenomenon of classification being easier than regression and saw the existence of regimes where classification works but regression does not. Does this phenomenon also manifest in overparameterized deep neural networks? The community has been working towards bridging the gap between theoretical understanding of linear models and deep networks and we conclude by providing a brief overview of key works in this area.

The key question that must be answered before we study generalization behavior of neural networks is what solution does an overparameterized neural network converge to based on choice of training algorithm and architecture. One line of work [66, 43, 2, 4, 31, 161] shows that wide neural networks, when initialized appropriately and trained using gradient descent behave like linear models and their weights don't change much from their initial values through the course of training. In this kind of lazy training regime, gradient descent learns the minimum-norm interpolator with respect to a particular kernel, namely the neural tangent kernel or NTK for short. Several works study generalization and benign overfitting for neural networks leveraging the NTK behavior[89, 102].

However, in practice can the weights of a network traverse far enough from their initialization such that the NTK approximation no longer holds [47]. Another line of work uses a statistical mechanics inspired mean-field approximation to study infinite-width networks by connecting the trajectory of the weights of the neural network when trained by stochastic gradient descent to the solution of a partial differential equation in a distribution space [30, 130, 96, 28]. By solving this differential equation we can determine what the network converges to.

How do networks behave in "rich" training regimes where they learn useful features during the course of training? This setting is much harder to analyze. For classification problems, the implicit bias of gradient descent on vanishing losses like logistic or cross-entropy loss or exponential losses have been studied in linear networks [58, 67] and non-linear networks [90, 106, 29] under various assumptions. In most settings the network converges to a max-margin (or equivalently minimum-norm) solution in some appropriate space. This regime is arguably more closely related to the performance of practical neural networks but, as [103] show, reaching this regime requires unrealistically small loss values, even in toy problems. The implicit bias of gradient-flow (gradient descent with infinitesimal step size) on networks where weights from different layers are initialized at different scales is studied in [7] and depending on the relative scales of initialization the implicit bias differs and one particular case of a two-layer neural network where the solution can be characterized explicitly is analyzed from a generalization perspective in [26].

Another well studied setting is the two layer neural network where only one layer is trained and the other is fixed (typically sampled from a random distribution) and doesn't change while training. Either the first layer is fixed during training and consists of random features [59, 88] or the second layer is fixed during training while the first layer is trained [48, 49, 23].

Understanding the implicit bias in more realistic and practically relevant regimes remains challenging in non-linear models with finite width where more than one layer is learned during training.

Bibliography

[1] Iosif Pinelis (https://mathoverflow.net/users/36721/iosif-pinelis). *Concentration and anti-concentration of gap between largest and second largest value in Gaussian iid sample*. MathOverflow. URL:https://mathoverflow.net/q/379688 (version: 2020-12-25). eprint: `https://mathoverflow.net/q/379688`. URL: `https://mathoverflow.net/q/379688`.

[2] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. "A convergence theory for deep learning via over-parameterization". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 242–252.

[3] Erin Allwein, Robert E. Schapire, and Yoram Singer. "Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers". In: *Journal of Machine Learning Research* 1 (2000), pp. 113–141.

[4] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. "On exact computation with an infinitely wide neural net". In: *Advances in Neural Information Processing Systems* 32 (2019).

[5] M. Athans, R. Ku, and S. Gershwin. "The uncertainty threshold principle: some fundamental limitations of optimal decision making under dynamic uncertainty". In: *IEEE Transactions on Automatic Control* 22.3 (1977), pp. 491–495.

[6] Navid Azizan, Sahin Lale, and Babak Hassibi. "A Study of Generalization of Stochastic Mirror Descent Algorithms on Overparameterized Nonlinear Models". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 3132–3136.

[7] Shahar Azulay, Edward Moroshko, Mor Shpigel Nacson, Blake E Woodworth, Nathan Srebro, Amir Globerson, and Daniel Soudry. "On the implicit bias of initialization shape: Beyond infinitesimal mirror descent". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 468–477.

[8] M. Baglietto, T. Parisini, and R. Zoppoli. "Numerical solutions to the Witsenhausen counterexample by approximating networks". In: *IEEE Transactions on Automatic Control* 46.9 (Sept. 2001), pp. 1471–1477. ISSN: 2334-3303. DOI: `10.1109/9.948480`.

[9] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. "Convexity, classification, and risk bounds". In: *Journal of the American Statistical Association* 101.473 (2006), pp. 138–156.

[10] Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. "Benign overfitting in linear regression". In: *Proceedings of the National Academy of Sciences* 117.48 (2020), pp. 30063–30070.

[11] Peter L Bartlett, Andrea Montanari, and Alexander Rakhlin. "Deep learning: a statistical viewpoint". In: *Acta numerica* 30 (2021), pp. 87–201.

[12] Andrew G Barto. "Connectionist Learning for Control: an overview". In: (1989).

[13] Mikhail Belkin. "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation". In: *Acta Numerica* 30 (2021), pp. 203–248.

[14] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. "Reconciling modern machine-learning practice and the classical bias–variance trade-off". In: *Proceedings of the National Academy of Sciences* 116.32 (2019), pp. 15849–15854.

[15] Mikhail Belkin, Daniel Hsu, and Ji Xu. "Two models of double descent for weak features". In: *SIAM Journal on Mathematics of Data Science* 2.4 (2020), pp. 1167–1180.

[16] Mikhail Belkin, Siyuan Ma, and Soumik Mandal. "To understand deep learning we need to understand kernel learning". In: *ICML* (2018).

[17] Mikhail Belkin, Alexander Rakhlin, and Alexandre B Tsybakov. "Does data interpolation contradict statistical optimality?" In: *arXiv preprint arXiv:1806.09471* (2018).

[18] Shai Ben-David, David Loker, Nathan Srebro, and Karthik Sridharan. "Minimizing the misclassification error rate using a surrogate convex loss". In: *Proceedings of the 29th International Coference on International Conference on Machine Learning*. 2012, pp. 83–90.

[19] Koby Bibas, Yaniv Fogel, and Meir Feder. "A New Look at an Old Problem: A Universal Learning Approach to Linear Regression". In: *CoRR* abs/1905.04708 (2019). arXiv: 1905.04708. URL: http://arxiv.org/abs/1905.04708.

[20] Anna Bosman, Andries Engelbrecht, and Mardé Helbig. "Visualising Basins of Attraction for the Cross-Entropy and the Squared Error Neural Network Loss Functions". In: *Neurocomputing* 400 (Mar. 2020). DOI: 10.1016/j.neucom.2020.02.113.

[21] Erin J. Bredensteiner and Kristin P. Bennett. "Multicategory Classification by Support Vector Machines". In: *Computational Optimization and Applications* 12 (1999), pp. 53–79. DOI: 10.1023/A:1008663629662.

[22] Roger W Brockett and Daniel Liberzon. "Quantized feedback stabilization of linear systems". In: *IEEE Transactions on Automatic Control* 45.7 (2000), pp. 1279–1289.

[23] Yuan Cao, Zixiang Chen, Mikhail Belkin, and Quanquan Gu. "Benign Overfitting in Two-layer Convolutional Neural Networks". In: *arXiv preprint arXiv:2202.06526* (2022).

[24] Horen Chang. "Presampling filtering, sampling and quantization effects on the digital matched filter performance". In: *Proceedings of the International Telemetering Conference*. 1982, pp. 889–915.

[25] Niladri S Chatterji and Philip M Long. "Finite-sample analysis of interpolating linear classifiers in the overparameterized regime". In: *Journal of Machine Learning Research* 22.129 (2021), pp. 1–30.

[26] Niladri S Chatterji, Philip M Long, and Peter L Bartlett. "The Interplay Between Implicit Bias and Benign Overfitting in Two-Layer Linear Networks". In: *arXiv preprint arXiv:2108.11489* (2021).

[27] Di-Rong Chen and Tao Sun. "Consistency of Multiclass Empirical Risk Minimization Methods Based on Convex Loss". In: *Journal of Machine Learning Research* 7 (Dec. 2006), pp. 2435–2447. ISSN: 1532-4435.

[28] Zixiang Chen, Yuan Cao, Quanquan Gu, and Tong Zhang. "A generalized neural tangent kernel analysis for two-layer neural networks". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 13363–13373.

[29] Lenaic Chizat and Francis Bach. "Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss". In: *Conference on Learning Theory*. PMLR. 2020, pp. 1305–1338.

[30] Lenaic Chizat and Francis Bach. "On the global convergence of gradient descent for over-parameterized models using optimal transport". In: *Advances in neural information processing systems* 31 (2018).

[31] Lenaic Chizat, Edouard Oyallon, and Francis Bach. "On lazy training in differentiable programming". In: *Advances in Neural Information Processing Systems* 32 (2019).

[32] Anna Choromanska, Alekh Agarwal, and John Langford. "Extreme multi class classification". In: *NIPS Workshop: eXtreme Classification, submitted*. Vol. 1. 2013, pp. 2–1.

[33] Chiranjib Choudhuri and Urbashi Mitra. "On Witsenhausen's counterexample: the asymptotic vector case". In: *2012 IEEE Information Theory Workshop, ITW 2012* (Sept. 2012), pp. 162–166. DOI: 10.1109/ITW.2012.6404649.

[34] Corinna Cortes, Vitaly Kuznetsov, Mehryar Mohri, and Scott Yang. "Structured Prediction Theory Based on Factor Graph Complexity". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett. Vol. 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper/2016/file/535ab76633d94208236a2e829ea6d888-Paper.pdf.

[35] Max Costa. "Writing on dirty paper (corresp.)" In: *IEEE transactions on information theory* 29.3 (1983), pp. 439–441.

[36] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, July 2006. ISBN: 0471241954.

[37] Koby Crammer, Yoram Singer, Nello Cristianini, John Shawe-taylor, and Bob Williamson. "On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines". In: *Journal of Machine Learning Research* 2 (2001), pp. 265–292.

[38] Yehuda Dar, Vidya Muthukumar, and Richard G Baraniuk. "A farewell to the bias-variance tradeoff? an overview of the theory of overparameterized machine learning". In: *arXiv preprint arXiv:2109.02355* (2021).

[39] Ahmet Demirkaya, Jiasi Chen, and Samet Oymak. "Exploring the Role of Loss Functions in Multiclass Classification". In: *2020 54th Annual Conference on Information Sciences and Systems (CISS)*. 2020, pp. 1–5. DOI: 10.1109/CISS48834.2020.1570627167.

[40] Zeyu Deng, Abla Kammoun, and Christos Thrampoulidis. "A model of double descent for high-dimensional binary linear classification". In: *Information and Inference: A Journal of the IMA* (Apr. 2021). iaab002. ISSN: 2049-8772. DOI: 10.1093/imaiai/iaab002. eprint: https://academic.oup.com/imaiai/advance-article-pdf/doi/10.1093/imaiai/iaab002/36872804/iaab002.pdf. URL: https://doi.org/10.1093/imaiai/iaab002.

[41] Thomas G Dietterich and Ghulum Bakiri. "Solving Multiclass Learning Problems via Error-Correcting Output Codes". In: *Journal of Artificial Intelligence Research* 2.1 (1994), pp. 263–286. ISSN: 1076-9757.

[42] Jian Ding, Yuval Peres, Gireeja Ranade, Alex Zhai, et al. "When multiplicative noise stymies control". In: *The Annals of Applied Probability* 29.4 (2019), pp. 1963–1992.

[43] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. "Gradient descent provably optimizes over-parameterized neural networks". In: *arXiv preprint arXiv:1810.02054* (2018).

[44] Nicola Elia and Jeff N Eisenbeis. "Limitations of linear remote control over packet drop networks". In: *Decision and Control (CDC), IEEE Conference on*. Vol. 5. IEEE. 2004, pp. 5152–5157.

[45] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*. Vol. 375. Springer Science & Business Media, 1996.

[46] Song Fang, Jie Chen, and Hideaki Ishii. *Towards Integrating Control and Information Theories*. Springer, 2016.

[47] Stanislav Fort, Gintare Karolina Dziugaite, Mansheej Paul, Sepideh Kharaghani, Daniel M Roy, and Surya Ganguli. "Deep learning versus kernel learning: an empirical study of loss landscape geometry and the time evolution of the neural tangent kernel". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 5850–5861.

[48] Spencer Frei, Yuan Cao, and Quanquan Gu. "Provable generalization of sgd-trained neural networks of any width in the presence of adversarial label noise". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 3427–3438.

[49] Spencer Frei, Niladri S Chatterji, and Peter L Bartlett. "Benign overfitting without linearity: Neural network classifiers trained by gradient descent for noisy linear data". In: *arXiv preprint arXiv:2202.05928* (2022).

[50] Jerome H Friedman. "On bias, variance, 0/1—loss, and the curse-of-dimensionality". In: *Data mining and knowledge discovery* 1.1 (1997), pp. 55–77.

[51] Johannes Fürnkranz. "Round Robin Classification". In: *Journal of Machine Learning Research* 2 (2002), pp. 721–747.

[52] Krzysztof Gajowniczek, Leszek Chmielewski, Arkadiusz Orłowski, and Tomasz Ząbkowski. "Generalized Entropy Cost Function in Neural Networks". In: *International Conference on Artificial Neural Networks*. Oct. 2017, pp. 128–136. ISBN: 978-3-319-68611-0. DOI: 10.1007/978-3-319-68612-7_15.

[53] Mario Geiger, Stefano Spigler, Stéphane d'Ascoli, Levent Sagun, Marco Baity-Jesi, Giulio Biroli, and Matthieu Wyart. "Jamming transition as a paradigm to understand the loss landscape of deep neural networks". In: *Physical Review E* 100.1 (2019), p. 012115.

[54] Pulkit Grover, Se Yong Park, and Anant Sahai. "Approximately Optimal Solutions to the Finite-Dimensional Witsenhausen Counterexample." In: *IEEE Trans. Automat. Contr.* 58.9 (2013), pp. 2189–2204.

[55] Pulkit Grover and Anant Sahai. "Witsenhausen's counterexample as assisted interference suppression". In: *International Journal of Systems, Control and Communications* 2.1-3 (2010), pp. 197–237.

[56] Yann Guermeur. "Combining Discriminant Models with New Multi-Class SVMs". In: *Pattern Anal. Appl.* 5 (June 2002), pp. 168–179. DOI: 10.1007/s100440200015.

[57] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. "Characterizing Implicit Bias in Terms of Optimization Geometry". In: *International Conference on Machine Learning*. 2018, pp. 1832–1841.

[58] Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. "Implicit bias of gradient descent on linear convolutional networks". In: *Advances in Neural Information Processing Systems* 31 (2018).

[59] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. "Surprises in high-dimensional ridgeless least squares interpolation". In: *arXiv preprint arXiv:1903.08560* (2019).

[60] Joo P Hespanha, Payam Naghshtabrizi, and Yonggang Xu. "A survey of recent results in networked control systems". In: *Proceedings of the IEEE* 95.1 (2007), pp. 138–162.

[61] Le Hou, Chen-Ping Yu, and Dimitris Samaras. "Squared Earth Mover's Distance-based Loss for Training Deep Neural Networks". In: *arXiv e-prints*, arXiv:1611.05916 (Nov. 2016), arXiv:1611.05916. arXiv: `1611.05916 [cs.CV]`.

[62] Daniel Hsu, Vidya Muthukumar, and Ji Xu. "On the proliferation of support vectors in high dimensions". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 91–99.

[63] Like Hui and Mikhail Belkin. "Evaluation of Neural Architectures Trained with Square Loss vs Cross-Entropy in Classification Tasks". In: *arXiv e-prints*, arXiv:2006.07322 (June 2020), arXiv:2006.07322. arXiv: `2006.07322 [cs.LG]`.

[64] Robert A Jacobs and Michael I Jordan. "Learning piecewise control strategies in a modular neural network architecture". In: *IEEE Transactions on Systems, Man, and Cybernetics* 23.2 (1993), pp. 337–345.

[65] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, Geoffrey E Hinton, et al. "Adaptive mixtures of local experts." In: *Neural Computation* 3.1 (1991), pp. 79–87.

[66] Arthur Jacot, Franck Gabriel, and Clément Hongler. "Neural tangent kernel: Convergence and generalization in neural networks". In: *Advances in neural information processing systems* 31 (2018).

[67] Ziwei Ji and Matus Telgarsky. "Gradient descent aligns the layers of deep linear networks". In: *arXiv preprint arXiv:1810.02032* (2018).

[68] Ziwei Ji and Matus Telgarsky. "The implicit bias of gradient descent on nonseparable data". In: *Conference on Learning Theory*. 2019, pp. 1772–1798.

[69] Abla Kammoun and Mohamed-Slim AlouiniFellow. "On the precise error analysis of support vector machines". In: *IEEE Open Journal of Signal Processing* 2 (2021), pp. 99–118.

[70] Johannes Karlsson, Ather Gattami, Tobias J Oechtering, and Mikael Skoglund. "Iterative source-channel coding approach to Witsenhausen's counterexample". In: *American Control Conference (ACC), 2011*. IEEE. 2011, pp. 5348–5353.

[71] Hyeji Kim, Yihan Jiang, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. "Deepcode: feedback codes via deep learning". In: *IEEE Journal on Selected Areas in Information Theory* PP (Apr. 2020), pp. 1–1. DOI: `10.1109/JSAIT.2020.2986752`.

[72] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *ICLR (Poster)*. 2015. URL: `http://arxiv.org/abs/1412.6980`.

[73] Ganesh Ramachandra Kini and Christos Thrampoulidis. "Analytic study of double descent in binary classification: The impact of loss". In: *2020 IEEE International Symposium on Information Theory (ISIT)*. IEEE. 2020, pp. 2527–2532.

[74] Doug M. Kline and Victor L. Berardi. "Revisiting squared-error and cross-entropy functions for training neural network classifiers". In: *Neural Computing & Applications* 14 (2005), pp. 310–318.

[75] Dmitry Kobak, Jonathan Lomond, and Benoit Sanchez. "The Optimal Ridge Penalty for Real-world High-dimensional Data Can Be Zero or Negative due to the Implicit Ridge Regularization." In: *Journal of Machine Learning Research* 21 (2020), pp. 169–1.

[76] Vladimir Koltchinskii and Dmitry Panchenko. "Empirical Margin Distributions and Bounding the Generalization Error of Combined Classifiers". In: *The Annals of Statistics* 30.1 (2002), pp. 1–50. ISSN: 00905364. URL: http://www.jstor.org/stable/2700001.

[77] Himanshu Kumar and P. Shanti Sastry. "Robust Loss Functions for Learning Multi-class Classifiers". In: *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (2018), pp. 687–692.

[78] Vitaly Kuznetsov, Mehryar Mohri, and Umar Syed. "Multi-Class Deep Boosting". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger. Vol. 27. Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper/2014/file/7bb060764a818184ebb1cc0d43d382aa-Paper.pdf.

[79] Vitaly Kuznetsov, Mehryar Mohri, and Umar Syed. "Rademacher Complexity Margin Bounds for Learning with a Large Number of Classes". In: *ICML Workshop on Extreme Classification: Learning with a Very Large Number of Labels*. 2015.

[80] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), p. 436.

[81] Jonathan T Lee, Edward Lau, and Yu-Chi Ho. "The Witsenhausen counterexample: A hierarchical search approach for nonconvex optimization problems". In: *IEEE Transactions on Automatic Control* 46.3 (2001), pp. 382–397.

[82] Yoonkyung Lee, Yi Lin, and Grace Wahba. "Multicategory Support Vector Machines: Theory and Application to the Classification of Microarray Data and Satellite Radiance Data". In: *Journal of the American Statistical Association* 99.465 (2004), pp. 67–81. DOI: 10.1198/016214504000000098. eprint: https://doi.org/10.1198/016214504000000098. URL: https://doi.org/10.1198/016214504000000098.

[83] Yunwen Lei, Urun Dogan, Alexander Binder, and Marius Kloft. "Multi-class SVMs: From Tighter Data-Dependent Generalization Bounds to Novel Algorithms". In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc., 2015.

[84] Yunwen Lei, Ürün Dogan, Ding-Xuan Zhou, and Marius Kloft. "Data-Dependent Generalization Bounds for Multi-Class Classification". In: *IEEE Transactions on Information Theory* 65.5 (2019), pp. 2995–3021. DOI: 10.1109/TIT.2019.2893916.

[85] Jian Li, Yong Liu, Rong Yin, Hua Zhang, Lizhong Ding, and Weiping Wang. "Multi-Class Learning: From Theory to Algorithm". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018.

[86] Na Li, Jason R Marden, and Jeff S Shamma. "Learning approaches to the Witsenhausen counterexample from a view of potential games". In: *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*. IEEE. 2009, pp. 157–162.

[87] Yue Li and Yuting Wei. "Minimum ℓ_1-norm interpolators: Precise asymptotics and multiple descent". In: *arXiv preprint arXiv:2110.09502* (2021).

[88] Zhu Li, Zhi-Hua Zhou, and Arthur Gretton. "Towards an understanding of benign overfitting in neural networks". In: *arXiv preprint arXiv:2106.03212* (2021).

[89] Tengyuan Liang, Alexander Rakhlin, and Xiyu Zhai. "On the multiple descent of minimum-norm interpolants and restricted lower isometry of kernels". In: *Conference on Learning Theory*. PMLR. 2020, pp. 2683–2711.

[90] Kaifeng Lyu and Jian Li. "Gradient descent maximizes the margin of homogeneous neural networks". In: *arXiv preprint arXiv:1906.05890* (2019).

[91] Yasaman Mahdaviyeh and Zacharie Naulet. "Risk of the Least Squares Minimum Norm Estimator under the Spike Covariance Model". In: *arXiv* (2019), arXiv–1912.

[92] Alexey S Matveev and Andrey V Savkin. *Estimation and Control Over Communication Networks*. Springer Science & Business Media, 2009.

[93] Andreas Maurer. "A vector-contraction inequality for rademacher complexities". In: *Algorithmic Learning Theory*. Ed. by Hans Ulrich Simon Ronald Ortner and Sandra Zilles. Springer International Publishing, 2016, pp. 3–17.

[94] William M McEneaney and Seung Hak Han. "Optimization formulation and monotonic solution method for the Witsenhausen problem". In: *Automatica* 55 (2015), pp. 55–65.

[95] Mustafa Mehmetoglu, Emrah Akyol, and Kenneth Rose. "A Deterministic Annealing Optimization Approach for Witsenhausen's and Related Decentralized Control Settings". In: *arXiv preprint arXiv:1403.5315* (2014).

[96] Song Mei, Theodor Misiakiewicz, and Andrea Montanari. "Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit". In: *Conference on Learning Theory*. PMLR. 2019, pp. 2388–2464.

[97] Song Mei and Andrea Montanari. "The generalization error of random features regression: Precise asymptotics and double descent curve". In: *arXiv preprint arXiv:1908.05355* (2019).

[98] Alan Miller. *Subset selection in regression*. Chapman and Hall/CRC, 2002.

[99] Partha P Mitra. "Understanding overfitting peaks in generalization error: Analytical risk curves for ℓ_2 and ℓ_1 penalized interpolation". In: *arXiv preprint arXiv:1906.03667* (2019).

[100] S. Mitter and A. Sahai. "Information and control: Witsenhausen revisited". In: *Lecture Notes in Control and Information Sciences* (1998), pp. 281–293.

[101] Andrea Montanari, Feng Ruan, Youngtak Sohn, and Jun Yan. "The generalization error of max-margin linear classifiers: High-dimensional asymptotics in the overparametrized regime". In: *arXiv preprint arXiv:1911.01544* (2019).

[102] Andrea Montanari and Yiqiao Zhong. "The interpolation phase transition in neural networks: Memorization and generalization under lazy training". In: *arXiv preprint arXiv:2007.12826* (2020).

[103] Edward Moroshko, Blake E Woodworth, Suriya Gunasekar, Jason D Lee, Nati Srebro, and Daniel Soudry. "Implicit bias in deep linear classification: Initialization scale vs training accuracy". In: *Advances in neural information processing systems* 33 (2020), pp. 22182–22193.

[104] Vidya Muthukumar, Adhyyan Narang, Vignesh Subramanian, Mikhail Belkin, Daniel J. Hsu, and Anant Sahai. "Classification vs regression in overparameterized regimes: Does the loss function matter?" In: *Journal of Machine Learning Research* 22 (2021), 222:1–222:69.

[105] Vidya Muthukumar, Kailas Vodrahalli, Vignesh Subramanian, and Anant Sahai. "Harmless interpolation of noisy data in regression". In: *IEEE Journal on Selected Areas in Information Theory* 1.1 (2020), pp. 67–83.

[106] Mor Shpigel Nacson, Suriya Gunasekar, Jason Lee, Nathan Srebro, and Daniel Soudry. "Lexicographic and depth-sensitive margins in homogeneous and non-homogeneous deep models". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 4683–4692.

[107] Mor Shpigel Nacson, Jason Lee, Suriya Gunasekar, Pedro Henrique Pamplona Savarese, Nathan Srebro, and Daniel Soudry. "Convergence of Gradient Descent on Separable Data". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. 2019, pp. 3420–3428.

[108] Girish N Nair and Robin J Evans. "Stabilization with data-rate-limited feedback: tightest attainable bounds". In: *Systems & Control Letters*. 41.1 (2000), pp. 49–56.

[109] Preetum Nakkiran. "More Data Can Hurt for Linear Regression: Sample-wise Double Descent". In: *arXiv e-prints*, arXiv:1912.07242 (Dec. 2019), arXiv:1912.07242. arXiv: 1912.07242 [stat.ML].

[110] Adhyyan Narang, Vidya Muthukumar, and Anant Sahai. "Classification and Adversarial examples in an Overparameterized Linear Model: A Signal Processing Perspective". In: *arXiv preprint arXiv:2109.13215* (2021).

[111] Kumpati S Narendra and Kannan Parthasarathy. "Identification and control of dynamical systems using neural networks". In: *IEEE Transactions on Neural Networks* 1.1 (1990), pp. 4–27.

[112] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. "In search of the real inductive bias: On the role of implicit regularization in deep learning". In: *arXiv preprint arXiv:1412.6614* (2014).

[113] Se Yong Park. "Information flow in linear systems". PhD thesis. UC Berkeley, 2013.

[114] Se Yong Park and Anant Sahai. "It may be easier to approximate decentralized infinite-horizon LQG problems". In: *Decision and Control (CDC), IEEE Conference on.* 2012, pp. 2250–2255.

[115] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. "Automatic differentiation in pytorch". In: *NIPS 2017 Autodiff Workshop: The Future of Gradient-based Machine Learning Software and Techniques* (2017).

[116] Bernardo Ávila Pires, Mohammad Ghavamzadeh, and Csaba Szepesvári. "Cost-Sensitive Multiclass Classification Risk Bounds". In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. Atlanta, GA, USA, 2013, pp. 1391–1399.

[117] Bernardo Ávila Pires and Csaba Szepesvári. "Multiclass Classification Calibration Functions". In: *arXiv e-prints*, arXiv:1609.06385 (Sept. 2016), arXiv:1609.06385. arXiv: 1609.06385 [stat.ML].

[118] Gireeja Ranade and Anant Sahai. "Control capacity". In: *IEEE Transactions on Information Theory* 65.1 (2018), pp. 235–254.

[119] Gireeja Ranade and Anant Sahai. "Non-Coherence in Estimation and Control". In: *51st Annual Allerton Conf. on Comm., Control, and Comp.* 2013.

[120] Ankit Singh Rawat, Jiecao Chen, Felix Xinnan X Yu, Ananda Theertha Suresh, and Sanjiv Kumar. "Sampled softmax with random fourier features". In: *Advances in Neural Information Processing Systems* 32 (2019).

[121] Dominic Richards, Jaouad Mourtada, and Lorenzo Rosasco. "Asymptotics of ridge (less) regression under general source condition". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 3889–3897.

[122] Ryan Rifkin and Aldebaro Klautau. "In Defense of One-Vs-All Classification". In: *Journal of Machine Learning Research* 5 (Dec. 2004), pp. 101–141.

[123] Ryan Michael Rifkin. "Everything old is new again: a fresh look at historical approaches in machine learning". PhD thesis. Massachusetts Institute of Technology, 2002.

[124] Mark Rudelson and Roman Vershynin. "Hanson-Wright inequality and sub-gaussian concentration". In: *Electronic Communications in Probability* 18 (2013).

[125] Fariborz Salehi, Ehsan Abbasi, and Babak Hassibi. "The impact of regularization on high-dimensional logistic regression". In: *Advances in Neural Information Processing Systems* 32 (2019).

[126] Luca Schenato, Bruno Sinopoli, Massimo Franceschetti, Kameshwar Poolla, and S Shankar Sastry. "Foundations of control and estimation over lossy networks". In: *Proceedings of the IEEE* 95.1 (2007), pp. 163–187.

[127] Vatsal Shah, Anastasios Kyrillidis, and Sujay Sanghavi. "Minimum norm solutions do not always generalize well for over-parameterized problems". In: *arXiv preprint arXiv:1811.07055* (2018).

[128] Alex Sherstinsky. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network". In: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306.

[129] Bruno Sinopoli, Luca Schenato, Massimo Franceschetti, Kameshwar Poolla, Michael I Jordan, and Shankar S Sastry. "Kalman filtering with intermittent observations". In: *Automatic Control, IEEE Transactions on.* 49.9 (2004), pp. 1453–1464.

[130] Mei Song, Andrea Montanari, and P Nguyen. "A mean field view of the landscape of two-layers neural networks". In: *Proceedings of the National Academy of Sciences* 115.33 (2018), E7665–E7671.

[131] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. "The implicit bias of gradient descent on separable data". In: *Journal of Machine Learning Research* 19.1 (2018), pp. 2822–2878.

[132] V. Subramanian, L. Brink, N. Jain, K. Vodrahalli, A. Jalan, N. Shinde, and A. Sahai. "Some new numeric results concerning the Witsenhausen counterexample". In: *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. Oct. 2018, pp. 413–420. DOI: 10.1109/ALLERTON.2018.8635853.

[133] Vignesh Subramanian, Rahul Arya, and Anant Sahai. *Generalization for multiclass classification with overparameterized linear models*. 2022. DOI: 10.48550/ARXIV.2206.01399. URL: https://arxiv.org/abs/2206.01399.

[134] Vignesh Subramanian, Moses Won, and Gireeja Ranade. "Learning a Neural-Network Controller for a Multiplicative Observation Noise System". In: *2020 IEEE International Symposium on Information Theory (ISIT)*. 2020, pp. 2849–2854. DOI: 10.1109/ISIT44484.2020.9174004.

[135] Hossein Taheri, Ramtin Pedarsani, and Christos Thrampoulidis. "Fundamental limits of ridge-regularized empirical risk minimization in high dimensions". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 2773–2781.

[136] Hossein Taheri, Ramtin Pedarsani, and Christos Thrampoulidis. "Sharp asymptotics and optimal performance for inference in binary models". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 3739–3749.

[137] S. Tatikonda and S. Mitter. "Control under communication constraints". In: *IEEE Transactions on Automatic Control* 49.7 (2004), pp. 1056–1068.

[138] Ambuj Tewari and Peter Bartlett. "On the Consistency of Multiclass Classification Methods." In: *Journal of Machine Learning Research* 8 (Jan. 2005), pp. 143–157.

[139] Christos Thrampoulidis, Samet Oymak, and Mahdi Soltanolkotabi. "Theoretical Insights Into Multiclass Classification: A High-dimensional Asymptotic View". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 8907–8920. URL: https://proceedings.neurips.cc/paper/2020/file/6547884cea64550284728eb26b0947ef-Paper.pdf.

[140] Shih-Hao Tseng and Ao Tang. "A local search algorithm for the Witsenhausen's counterexample". In: *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE. 2017, pp. 5014–5019.

[141] Alexander Tsigler and Peter L Bartlett. "Benign overfitting in ridge regression". In: *arXiv preprint arXiv:2009.14286* (2020).

[142] George L. Turin. "An introduction to digitial matched filters". In: *Proceedings of the IEEE* 64.7 (1976), pp. 1092–1112.

[143] Vladimir Naumovich Vapnik. "An overview of statistical learning theory". In: *IEEE transactions on Neural Networks* 10.5 (1999), pp. 988–999.

[144] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*. Vol. 48. Cambridge University Press, 2019.

[145] Guillaume Wang, Konstantin Donhauser, and Fanny Yang. "Tight bounds for minimum l1-norm interpolation of noisy data". In: *arXiv preprint arXiv:2111.05987* (2021).

[146] Ke Wang, Vidya Muthukumar, and Christos Thrampoulidis. "Benign Overfitting in Multiclass Classification: All Roads Lead to Interpolation". In: *arXiv e-prints*, arXiv:2106.10865 (June 2021), arXiv:2106.10865. arXiv: 2106.10865 [stat.ML].

[147] Weichen Wang and Jianqing Fan. "Asymptotics of empirical eigenstructure for high dimensional spiked covariance". In: *Annals of statistics* 45.3 (2017), p. 1342.

[148] Jason Weston and Chris Watkins. *Multi-class Support Vector Machines*. Tech. rep. 1998.

[149] R. J. Williams and J. Peng. "An Efficient Gradient-Based Algorithm for On-Line Training of Recurrent Network Trajectories". In: *Neural Computation* 2.4 (Dec. 1990), pp. 490–501. ISSN: 0899-7667. DOI: 10.1162/neco.1990.2.4.490.

[150] Hans S Witsenhausen. "A counterexample in stochastic optimum control". In: *SIAM Journal on Control* 6.1 (1968), pp. 131–147.

[151] Blake Woodworth, Suriya Gunasekar, Jason D. Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. "Kernel and Rich Regimes in Overparametrized Models". In: *Proceedings of Thirty Third Conference on Learning Theory*. Ed. by Jacob Abernethy and Shivani Agarwal. Vol. 125. Proceedings of Machine Learning Research. PMLR, 2020, pp. 3635–3673. URL: https://proceedings.mlr.press/v125/woodworth20a.html.

[152] Denny Wu and Ji Xu. "On the Optimal Weighted ℓ_2 Regularization in Overparameterized Linear Regression". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 10112–10123.

[153] Jingfeng Wu, Difan Zou, Vladimir Braverman, and Quanquan Gu. "Direction matters: On the implicit bias of stochastic gradient descent with moderate learning rate". In: *arXiv preprint arXiv:2011.02538* (2020).

[154] Nan Xiao, Lihua Xie, and Li Qiu. "Feedback stabilization of discrete-time networked systems over fading channels". In: *IEEE Transactions on Automatic Control* 57.9 (2012), pp. 2176–2189.

[155] Liang Xu, Yilin Mo, Lihua Xie, and Nan Xiao. "Mean Square Stabilization of Linear Discrete-time Systems over Power Constrained Fading Channels". In: *IEEE Transactions on Automatic Control* (2017).

[156] Ian En-Hsu Yen, Xiangru Huang, Pradeep Ravikumar, Kai Zhong, and Inderjit Dhillon. "Pd-sparse: A primal and dual sparse approach to extreme multiclass and multilabel classification". In: *International conference on machine learning*. PMLR. 2016, pp. 3069–3077.

[157] Serdar Yuksel and T Basar. *Stochastic Networked Control Systems*. Springer, 2013.

[158] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. "Understanding deep learning requires rethinking generalization". In: *arXiv preprint arXiv:1611.03530* (2016).

[159] Tong Zhang. "Statistical Analysis of Some Multi-Category Large Margin Classification Methods". In: *Journal of Machine Learning Research* 5 (2004), pp. 1225–1251.

[160] Tong Zhang. "Statistical behavior and consistency of classification methods based on convex risk minimization". In: *Annals of Statistics* (2004), pp. 56–85.

[161] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. "Gradient descent optimizes over-parameterized deep ReLU networks". In: *Machine learning* 109.3 (2020), pp. 467–492.