Title
Statistical Learning for Sparse Sensing and Agile Operation

Author
Zhou, Yuxun

Publication Date
2017

Peer reviewed|Thesis/dissertation

Statistical Learning for Sparse Sensing and Agile Operation

by

Yuxun Zhou

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 Costas J. Spanos, Chair
Professor Peter L. Bartlett
Professor Stefano Schiavon

Spring 2017

Statistical Learning for Sparse Sensing and Agile Operation

Abstract

Statistical Learning for Sparse Sensing and Agile Operation

by

Yuxun Zhou

Doctor of Philosophy in Engineering – Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Costas J. Spanos, Chair

Recent advancements in the study of cyber-physical systems (CPS) have addressed the combination of computation, networking, physical processes, and human involvement as an overall system to improve adaptability, autonomy, efficiency, functionality, reliability, safety, and usability. Among other lines of CPS research, machine learning (ML) has emerged as an indispensable component for state estimation, prediction, diagnosis, structure identification, operation specification, event detection, etc. This work particularly discusses three learning tasks that are commonly encountered in CPS sensing and operation applications. The primary motivations underlying this dissertation are (1) to incorporate the unique characteristics of the data generated from system measurement, and (2) to facilitate the integration of ML into other components of CPS, such as sensing and control subsystems.

More specifically, we first consider learning interaction structures for sparse sensing. With the generic directed information maximization as the learning objective, we discuss two subset selection problems and provide performance guarantees for greedy algorithms by extending the notion of submodularity. Practically, the proposed learning framework can be applied broadly to streaming feature selection, causality mining, sensor placement, as well as the construction of causal graphs. The second learning task discussed in this work is focused on the detection of outliers or novelties from multiple correlated time series data generated from CPS measurement. The key issue being addressed is the utilization of the correlation information in the smoothing process of multiple sequences. Two methods, one based on a multi-task extension of non-parametric time series modeling and the other based on merging hidden Markov model with matrix factorization, are established and analyzed. Lastly, we discuss the task of learning system requirement for agile operation and optimal control. The classical ML paradigm is modified with "shape constraints" to facilitate its usage for optimal control or to capture class imbalance for event detection. While developing new learning formulations, we also propose a novel global optimization procedure, namely parametric dual maximization, that is able to solve a class of modified machine learning problems having non-convex objectives.

To Joshua Zhou.

Contents

List of Figures

List of Tables

Acknowledgments

First and Foremost, I would like to thank my adviser and mentor, Professor Costas J. Spanos, for guiding me through my graduate studies at Berkeley. He has given me both freedom and guidance to pursue my research interests in novel and interesting directions. I cannot thank him enough.

I would like to thank Professor Peter Bartlett for being my Qualifying Exam committee, dissertation committee and GSI adviser. He was an excellent mentor to me and I gained substantial knowledge and encouragement of the research and teaching through him.

Also I would like to thank Professors Stefano Schiavon and Alexandre Bayen for being on my Qualifying Exam committee and dissertation committee. They have given me very helpful feedback for my graduate work. I would like to thank all our group members: Jae Yeon Baek, Zhaoyi Kang, Ming Jin, Ruoxi Jia, Han Zou, Dan Li, Ioannis Konstantakopoulos, and Yovana Gomez for all their support and help.

Finally, I would like to express my deepest gratitude to my family for their love, compassion and support in my endeavor. My wife, Lu Ding, deserves to share my degree as had it not been for her unbounded love and constant support, my journey would not have even started. My parents are very supportive for my PhD. They tried very hard to shield me from the troubles happening thousands miles away at my hometown. Without the support and understanding from my family, I could not have gone so far. This is for them.

Chapter 1

Introduction and Motivation

1.1 Sensor Rich Cyber-Physical Systems

1.1.1 Background

Modern engineering systems feature a combination of computation, networking, and physical processes, and are tightly integrated with the demand and behavior of their users. Moreover, the incorporation of intelligent applications, ubiquitously connected usage patterns, and increased performance demands have changed the dynamics and interactions of cyber and physical components in a significant manner. These trends require fundamentally new modeling, design, and diagnosis approaches where both cyber and physical components are jointly considered at all levels of abstraction. Recent advancements in the research-field of cyber-physical systems (CPSs), have been addressing these issues to improve modern engineering systems in terms of adaptability, autonomy, efficiency, functionality, reliability, safety, and usability [1, 2, 3, 4, 5, 6]. The subjects of the CPS research include but are not limited to complex automotive and aviation systems, intelligent traffic scheduling and control, reliable medical devices, environmental monitoring and control systems, distributed robotics, electric power grid, communication systems, etc. [7, 8, 9, 10].

Among other defining characteristics of CPS, the impact of computational components on the other aspects of the system is most remarkable. Due to the accelerated advancement of sensing and measurement technology, the cyber components now have access to rich information related with the dynamics, states, and behavior of the physical components and their users. The computation enabled by those miscellaneous information sources, therefore, broadly involves many areas of artificial intelligence (AI) including machine learning (ML), information representation, scheduling, optimal control etc. This works specifically discusses several arising problems of machine learning as the indispensable cyber component of sensor rich CPS. To illustrate the necessity and effects of integrating learning algorithms into the overall system, we briefly introduce the smart building technology as an example.

1.1.2 An Example of CPS: the Smart Building Technology

Traditionally buildings are treated only as physical entities that provide services such as sheltering, security, living/working space, privacy, storage, comfort, culture and personal values, a form of investment, etc. The research lines addressing buildings are mainly from architectural, structural, and energy efficiency points of view. Recently with the unprecedented development of information technology and sensor network, buildings are becoming a complex combination of both physical and cyber subsystems. The newly integrated intelligent control and communication module, monitoring subsystem (e.g., sensor networks) and decision support system enable smart building technology that greatly expands the functionality and improves energy efficiency and well-beings of building occupants. The co-existence with the physical components, such as architectural structures, civil engineering infrastructure, heating ventilation and air conditioning (HVAC) systems, makes modern smart building a unique case of CPS. As of today, the energy consumption of buildings, both residential and commercial, accounts for over 40% of primary energy usage in the U.S [11]. With a novel CPS-based perspective for design, deployment and operation taken, it can be expected that much of this would be reduced. In addition, the security, privacy, comfort and productivity of building occupants can be greatly enhanced as new utilities are made available by leveraging sensing, prediction, and personalized control [12, 13, 7, 8, 14].

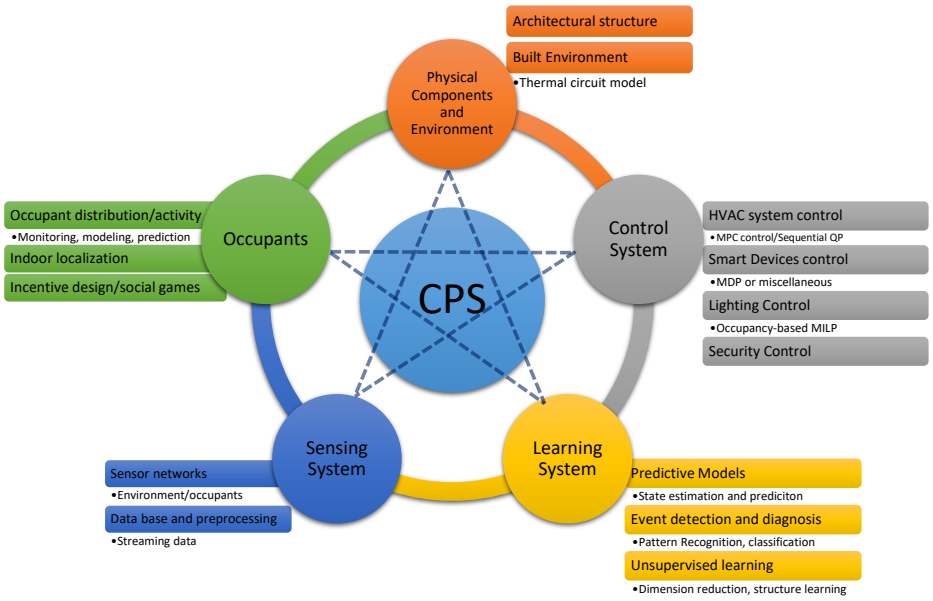


Figure 1.1: Smart Building as CPS: physical and cyber components

The major subsystems of a smart building are illustrated in Figure 1.1. Besides traditional components like structure environment and occupants, one physical subsystem (sensor networks), and two cyber components (control and ML algorithms), are also integrated in the overall CPS. All the five components are inherently coupled together and should be con-

sidered in a unified framework to realize new applications, improve performance, and reduce operational costs.

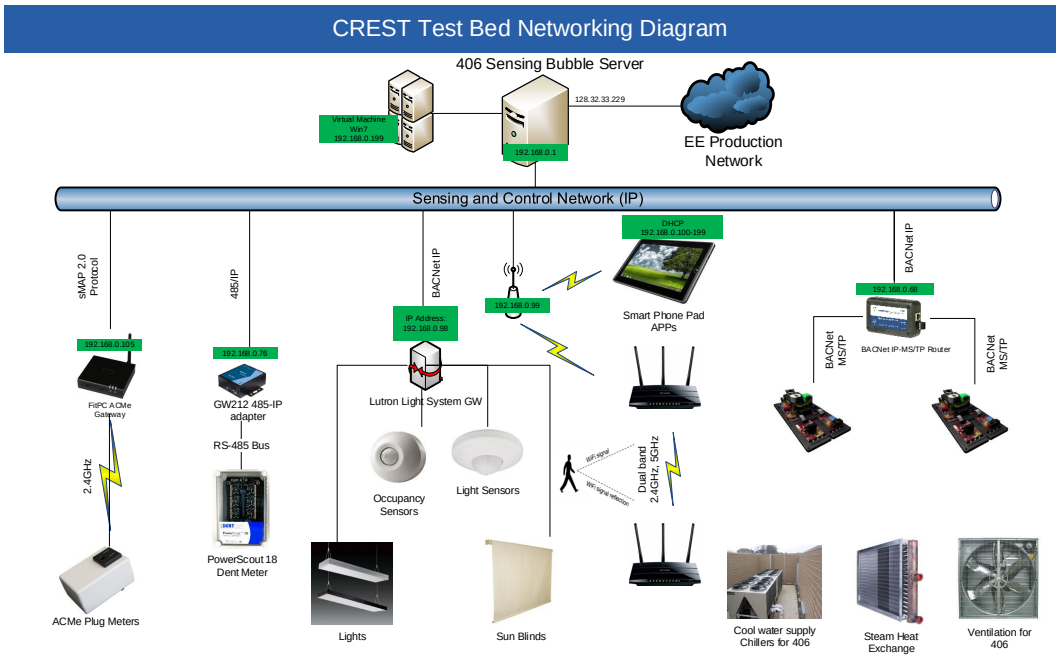


Figure 1.2: Sensor network deployment at CREST center, Cory Hall, Berkeley

The sensing system in smart buildings is devoted to monitoring built environment, occupancy, HVAC states, electricity, functionality of miscellaneous devices, etc. It consists not only of sensors and measurement instruments, but also of data base and communication networks for information storage and transmission. For a concrete example, Figure 1.2 shows the sensing system deployment in the Center for Research in Energy Systems Transformation (CREST) at Cory hall, UC Berkeley. The environmental variables of the working space, including temperature, humidity, light, air quality, CO_2 concentration, are monitored through the Building-in-Briefcase (BiB) sensing platform [7]. The energy usages of miscellaneous devices, such as desktops, laptops, printers, coffee machine, microwave, etc., are recorded using the AC meters (ACme) developed by UC Berkeley as part of the Green Soda Project [15]. In addition to device-level energy monitoring, the overall electricity profile is measured with the high resolution PowerScout 18 Dent Meter, which provides per second three phase voltage and current readings [16]. The radio-frequency identification (RFID) [17, 18], bluetooth [19] and WiFi based indoor localization system [20, 21, 22, 23] gather electromagnetic signals related to the presence and activities of the space occupants. Last but not least, the operation status of the Lutron light system and HVAC system is reported from integrated sensors to the building management database by using the Building Automation and Control Networks (BACnet) protocol [24].

As shown in Figure 1.2, all sensing data is eventually transmitted to the building management system (BMS) for processing and utilization. This is where machine learning and control algorithms come into play. The ML module takes the raw measurement as input, and performs a wide variety of tasks ranging from denoising, prediction, dimension reduction, to complex ones like state estimation, event classification, causal identification, etc [25, 26, 27]. To list a few in the CREST example, the device-level electricity data is processed with generalized principle component analysis (gPCA) [28], a multivariate autoregressive integrated moving average (mARIMA) model [29], as well as a directed information filter [30] for dimension reduction, prediction, and causal analysis, respectively. Given the environmental information, convex functions are learned to model users' comfort requirement [31], and a particle filter based approach is applied to estimate occupants' presence and activity [32, 33, 9]. The combination of sensing systems and ML provides rich information about the overall system state, behavior, faults and events that would otherwise be unobservable using traditional measurements and modeling techniques.

The control module in smart buildings mainly deals with HVAC and lighting systems. Recently a large body of research has been motivating a transition from the classic rule based control strategies to more comprehensive optimal control schemes. Regarding smart building technologies in particular, the adaptations of Model Predictive Control (MPC) [34] have achieved significant improvement in terms of both energy efficiency and demand response. Essentially, MPC treats building thermal space model as the control subject, uses the observed environmental/occupancy information from the ML system to guide the physical model, and finally optimizes over a receding horizon to reduce total and peak energy usage [31]. Relying largely on light sensors and occupancy information from the ML module, the optimal control of lighting system in smart buildings can be formulated as a mixed integer linear programming (MILP), which is solved with the branch and bound algorithm to provide control signals for each light [35]. In short, the control component takes the output of sensing system and ML algorithm as its input, and directly modulate the physical component to achieve the desired system behavior.

1.2 Machine Learning for Sensor Rich CPS

Previous research on system engineering was mainly focused on developing physical models of the underlying processes. The study of sensor rich CPS, however, calls for a combination of a model-based approach and a model-less data-driven approach for state estimation, structure identification, and control. The motivations are two-fold: On one hand, the physical model based methods largely rely on the correctness of the presumptive dynamics of the system. Their limitations are obvious as (1) Modern CPS integrates many heterogeneous, strongly coupled, and high dimensional components together, which are hard, if not impossible, to be described with simplified physical laws. (2) More and more CPS applications have to deal with significant randomness caused by human involvement, the chaotic nature of the system, or the unobservability of the process, which significantly deteriorate the reliability of physical

models. On the other hand, provided with rich sensing measurement of a CPS, it seems more appropriate to replace part of the modeling task with a data-driven approach, or even conduct pure machine learning based analysis for statistical inference and decision making. As both sensing technology and ML have been advanced greatly, the data-driven approach is expected to receive increasing attention in both application and research domains.

1.2.1 Learning Tasks

Following the above discussion, the machine tasks involved in CPS applications can be summarized into the following categories:

- State Estimation: The most common learning objective in CPS aims to infer system states that are not accessible or even unobservable with the deployed sensing measurement. Those "hidden states" could be parameters of a complex physical process that are hard to measure directly, or human involved factors that are inaccessible due to privacy or security concerns. In the dynamic modeling literature, tasks are usually referred to as smoothing, filtering, and prediction, depending on the temporal location of the parameter under estimation. From a broader machine learning perspective, however, the task of state estimation is essentially a statistical inference problem, which can be dealt with by a wide variety of parametric or non-parametric methods.

- Interaction Identification: Data available in CPS is mainly from the measurement of different parts of the system that are inherently interactive and coupled together. The task of interaction identification is to reveal the underlying relatedness of physical processes regarding their dependence and causality. Provided with the interaction structure, one is not only able to understand the CPS in a more compact, reduced dimensional space, but also incorporate that information to improve various estimation tasks through feature selection, transfer learning, collaborative filtering, etc.

- Learning System Specifications: An unique learning task in CPS is the modeling of system requirements or specifications needed for proper functionality and operation. In the smart building example, the occupants' thermal comfort, lighting, and acoustic requirement have to be learned from survey and sensing data for the operation of several subsystems. The energy consumption curves of the HVAC system have to be fitted using empirical measurement under different environmental conditions, so as to enable an energy efficient control for all scenarios. Generally speaking, modern CPS contains components that are either inherently uncertain or hard to describe with physical laws. Such components are better characterized by statistical and ML models which can be learned from rich measurement data.

1.2.2 Key Challenges

Although some of the learning tasks mentioned in the previous section may appear conventional and well-solved at first glance, we point out the following arising issues in the context of CPS, that call for the development of novel machine learning paradigms.

- Since a CPS is usually operated in multiple modes under different conditions, the resultant measurement data is often non-stationary and discontinuous. More importantly, the relatedness of multiple measurements is universally present and should be incorporated to establish temporal-spatial, multi-variate, or multi-task learning.
- Given multiple measurement data, the identification of interaction structure is naturally a combinatorial problem that is worst-case NP-hard. Besides, the interaction usually exhibits itself in the form of sequential influence or causality, which is hard to capture using traditional statistics. Hence a resort to approximation algorithms and a generalized dependence metric is required for interaction identification.
- As far as learning system specifications are concerned, the ML model has to take into account: (1) The characteristics of the measurement data, which is often corrupted, imbalanced, and lacks proper labels; (2) The modified learning objective, which is often cost-sensitive, needs robustness to data corruptions, and more importantly has to satisfy the requirements imposed by CPS monitoring, diagnosis, and control applications.

1.3 Thesis Outline

This work discusses the aforementioned learning tasks and proposes customized machine learning tools that address the above challenges. Although CPS are the general background for the application of the discussed methods, we present and derive those tools in a broader and more rigorous ML framework. The usage of the proposed methods is illustrated in the experiment part of each chapter with their source code provided for practical purposes. The rest of the thesis is organized as follows:

Chapter 2, entitled "Learning Causal Interactions for Sparse Sensing", is focused on variable selection and structure identification with an information theoretical metric that is able to capture more general dependence. Technically, the learning tasks are first related to subset selection problems, and then greedy approximations are studied as the preferred solution by extending the notion of submodularity. Practically, the results and methods proposed in this chapter can be readily used for feature selection with streaming measurement, causality mining for multiple system variables, sparse sensor placement, as well as the construction of dependency graph for the interaction of various CPS processes.

The following chapter discusses the detection of outliers or novelties from multiple correlated time series data. The key issue addressed here is the incorporation of the correlation information in the smoothing process of multiple time series. Two methods, one based on a multi-task extension of non-parametric time series model and the other by merging hidden

Markov model with matrix factorization, are established in this chapter. The applications to fault detection from CPS sensing data show that the proposed multi-task methods are able to reveal interesting outliers/novelties that might be ignored using traditional single task learning methods.

Chapter 4 is devoted to learning system requirement for agile operation and optimal control. We start by addressing the problem of building a piece-wise convex classifier to model system operation constraints, and then extend the classifier to a more general veto-consensus multiple kernel learning framework for fault detection, domain description, and semi-supervised event diagnosis. The technical contribution of this chapter is more toward optimization: we develop a novel global optimization procedure, namely parametric dual maximization (PDM), that is able to solve a class of modified machine learning problems having non-convex objectives. In the experiment part, we not only test the performance of PDM and show its advantage over state-of-the-art optimization methods, but also provide two case studies that demonstrate the usage of the proposed ML schemes for CPS optimal control and event detection applications.

Finally, Chapter 5 concludes this study and includes a brief discussion about future tasks found within the topics of this thesis.

Chapter 2

Learning Causal Interactions for Sparse Sensing

2.1 Introduction and Motivation

Recent advances in sensor network and information technologies have granted researchers access to large amounts of time series data. In the context of cyber physical systems (CPS) in particular, high-resolution measurements of the physical, cyber and human involved processes are accumulated to provide enhanced observability of the system, enabling application like machine learning, control, diagnosis, gamification, etc. Taking the smart building application for example, the environmental sensing technology measures the temperature, humidity, air quality (CO_2), sound pressure level, etc of the spaces of interest. The smart power meters are utilized to record the energy consumption of the heating, ventilation and air conditioning (HVAC) systems, the electricity usage of lighting, plug loads, as well as other ancillary apparatus in smart buildings. Moreover, the occupancy sensing platform, enabled by RFID [36, 37], Bluetooth [19], embedded sensors in mobile phones [38, 39, 40], environmental sensors [8] and WiFi [41, 42, 43, 44, 21], provides detailed information about individual presence, location, even behavior and activities [45, 22, 46]. Given rich information in the form of high dimensional time series data, the key issue is to obtain a concise or ideally sparse information representation for downstream applications such as prediction, event detection, system diagnosis, and control. Towards this goal, this chapter is focused on resolving the following problems that are related to sparse sensing and information representation:

- Sensor placement, i.e., deciding which information stream is worth measuring, such that the overall observability of the system is maximized.
- Covariate selection, i.e., picking up useful covariates, to benefit the prediction or estimation of a target process.

- Graphical representation of interactions, i.e., identifying a structure that captures the direct influence, possibly causal impact, among processes of interest.

We propose to use *Directed Information* as a measure of generic dependence and causality, and show that the aforementioned problems can be reduced to two fundamental subset selection problems. Both of them try to maximize cardinality constrained directed information. To attack the NP-hard subset seletion problems we resort to approximate algorithms. More specifically we study the performance of greedy heuristics through submodularity analysis. To handle the possible lack of submodularity, we introduce the *submodularity index (SmI)* as a key quantity to characterize the degree of submodularity for general set functions. Using the new index, stronger performance guarantee of greedy heuristics is found for submodular functions, significantly improving previous bound. More importantly, performance guarantee is obtained for possibly non-monotonic and non-submodular functions, extending greedy algorithms to the maximization of a much broader class of functions.

With regards to the subset selection objectives considered in this chapter, we provide detailed analysis of their SmI and make a connection between causal subset selection and causal graphs learning. Finally, an efficient structure learning algorithm is proposed to construct a sparse representation of the interaction from multiple time series data. The theoretical analysis and the structure learning algorithm are tested on both synthesis and real world data sets, and the results justify the effectiveness of the proposed solution.

The rest of the chapter is organized as follows. In next section, we briefly review the notion of directed information and submodular functions. Section 2.3 is devoted to the formulation of two causal subset selection problems and their submodularity analysis. In Section 2.4, we introduce SmI and provide an analysis of the random greedy algorithm. Following the obtained results, the method for causal graph structure learning is given in Section 2.5. Finally, the experimental results are presented in Section 2.6.

2.2 Preliminary: Subset Selection, Directed Information and Submodular Function

2.2.1 Subset Selection and Directed Information

A wide variety of research disciplines, including computer science, economics, biology, and various social sciences, involve causality analysis of a network of interacting random processes. In particular, many of those tasks are closely related to subset selection. For example, in social network research, it is critical for advertisers to target opinion leaders to maximize the influence of their messages. In stock market analysis, investors are interested in selecting causal covariates from a pool of data streams, in order to better predict the stock of interest. Likewise in sensor network applications, with a limited budget it is not only beneficial but also mandatory to optimally place sensors at information "sources" that provide the best observability of the system.

To solve the aforementioned problems we firstly need a causality measure for multiple random processes. In literature, there exists two types of causality definitions, one is related to time series prediction (called Granger-type causality [47]) and another with counter-factuals analysis [48]. Under the framework of Grange, one establishes a causal relation if the one time series contains *unique* information for the prediction of the other. Traditionally, with linear regression or other time series prediction models, Granger Causality can be reduced to a certain model selection problem, which is usually dealt with by hypotheis testing [49, 50]. The counter-factuals analysis tries to substantiate "a comparison between what actually happened and what would have happened in the absence of the intervention", which is a more intuitive procedure from a philosophical perspective. Practical algorithms in this catergory include Structural Equation Modeling (SEM) [51] and its non-Gaussian extension [52], etc.

In this work, we focus on Granger-type prediction causality substantiated with *Directed Information (DI)*, a tool from information theory. Recently, a large body of work has successfully employed DI in many research fields, including influence mining in gene networks [53], causal relationship inference in neural spike train recordings [54], and message transmission analysis in social media [55]. Compared to model-based or testing-based methods such as [49, 56], DI is not limited by model assumptions and can naturally capture non-linear and non-stationary dependences among random processes. In addition, it has clear information theoretical interpretation and admits well-established estimation techniques. In this regards, we formulate causal sensor placement and covariate selection as cardinality constrained directed information maximizations problems.

Now we formalize the definition of Directed Information. Consider two random process X^n and Y^n, we use the convention $X^i = \{X_0, X_1, ...X_i\}$, with $t = 0, 1, ..., n$ as time index. Directed Information from X^n to Y^n is defined in terms of mutual information:

$$\mathcal{I}(X^n \to Y^n) = \sum_{t=1}^{n} I(X^t; Y_t | Y^{t-1}) \tag{2.1}$$

which can be viewed as the aggregated dependence between the history of process X and current value of process Y, given past observations of Y. The above definition captures a natural intuition about causal relationship, i.e., the unique information X^t has on Y_t, when the past of Y^{t-1} is known. With the chain rule of entropy, directed information is usually written in the following form

$$\begin{aligned} \mathcal{I}(X^n \to Y^n) &= \sum_{t=1}^{n} \left\{ H(Y_t | Y^{t-1}) - H(Y_t | Y^{t-1}, X^t) \right\} \\ &= H(Y^n) - \sum_{t=1}^{n} H(Y_t | Y^{t-1}, X^t) \end{aligned} \tag{2.2}$$

The first line of (2.2) shows another intuition: Since entropy is a measure of uncertainty in bits, the directed information is actually the aggregated difference between uncertainty of Y

given its past history and the uncertainty but given additional information from the process X. The last term in (2.2) is usually referred to as *causally conditioned entropy*.

$$H(Y^n||X^n) \triangleq \sum_{t=1}^{n} H(Y_t|Y^{t-1}, X^t) \tag{2.3}$$

The directed information from X^n to Y^n when *causally conditioned* on the series Z^n is defined as

$$\begin{aligned} \mathcal{I}(X^n \to Y^n||Z^n) &= H(Y^n||Z^n) - H(Y^n||X^n, Z^n) \\ &= \sum_{t=1}^{n} I(X^t; Y_t|Y^{t-1}, Z^t) \end{aligned} \tag{2.4}$$

Observe that causally conditioned directed information is expressed as the difference between two causally conditioned entropy, which can be interpreted as "causal uncertainty reduction". With this one is able to relate directed information to Granger Causality. Denote $\bar{X}$ as the complement of X in a a universal set V. Then,

Theorem 1. *[57] With log loss, $\mathcal{I}(X^n \to Y^n||\bar{X}^t)$ is precisely the value of the side information (expected cumulative reduction in loss) that X has, when sequentially predicting Y with the knowledge of $\bar{X}$. The predictors are distributions with minimal expected loss.*

In particular, with linear models directed information is equivalent to Granger causality for jointly Gaussian processes. For stationary processes, the notion of information rate can be naturally extended to DI. Assuming sufficient condition for the existence of limits and enough regularity for the conditional probability measures, the causally conditioned entropy rate and directed information rate are defined by:

$$H(X||Y) = \lim_{n\to\infty} \frac{1}{n} H(Y^n||X^n) \tag{2.5}$$

$$\mathcal{I}(X \to Y) = \lim_{n\to\infty} \frac{1}{n} \mathcal{I}(X^n \to Y^n) \tag{2.6}$$

2.2.2 Submodular Function

In literature, the study of submodular functions for subset selection and other related machine learning problems has shown promising results in both theory and practice. Following the pioneer work [58, 59] that has proven the near optimal $1 - 1/e$ guarantee of greedy heuristics, [60, 61] investigates the submodularity of mutual information under Gaussian processes, and then uses a greedy algorithm for sensor placement. In the context of speech and nature language processing (NLP), [62, 63] adopted submodular objectives that encourage small vocabulary subset and large coverage, and then proceeded to maximization with a modified greedy heuristic. In [64], the authors combine insights from spectral analysis of

covariance and the submodularity of R^2 score. Remarkably, their result explains the near optimal performance of Forward Regression and Orthogonal Matching Pursuit methods.

There are three equivalent definitions of submodular functions, and each of them reveals a distinct interpretation of submodularity, a natural diminishing returns property that universally exists in economics, game theory and network systems.

Definition 1. *Submodular Set Function*
A submodular funciton is a set function $f : 2^{\Omega} \to \mathbb{R}$, which satisfies one of the three equivalent definitions:

1. *For every $S, T \subseteq \Omega$ with $S \subseteq T$, and every $x \in \Omega \setminus T$, we have that*

$$f(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T) \tag{2.7}$$

2. *For every $S, T \subseteq \Omega$, we have that*

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T) \tag{2.8}$$

3. *For every $S \subseteq \Omega$, and $x_1, x_2 \in \Omega \setminus S$, we have that*

$$f(S \cup \{x_1\}) + f(S \cup \{x_2\}) \geq f(S \cup \{x_1, x_2\}) + f(S) \tag{2.9}$$

A set function f is called *supermodular* if $-f$ is submodular. The first definition is direct related with the diminishing return property: The two sides of (2.7) can be thought of as marginal returns of the set function f at S versus the return at T, by adding an additional element x. The second definition is better understood as the classic max k-cover problem [65]. The third definition indicates that the contribution of two elements is maximized when they are added separately into the base set. Note that this property can be easily extended to the case with general k elements, which will be used later to define submodularity index.

View $f(S \cup \{x\}) - f(S)$ as a "first order derivative" of f at base set S, the first definition in fact requires non-increasing derivative. Consequently, submodularity appears to be similar to "concavity" for set functions. Throughout this paper, we will denote

$$f_X(S) \triangleq f(S \cup X) - f(S)$$

for further analysis. The "concavity" intuition coincides with the well known fact that, despite of being NP-hard, maximizing submodular functions with a simple greedy heuristic has near optimal performance guarantees [66]. On the other hand, it is worth pointing out that submodularity is also closely related to "convexity" due to the convex Lovász extension, with which polynomial time algorithms, such as [67] $O(n^5\alpha + n^6)$, can be designed for unconstrained minimization.

2.3 Formulation and Submodularity Analysis

2.3.1 Problem Formulation

In this section, we first formulate the causal subset selection problem into cardinality constrained directed information maximization. Depending on different scenarios, two objective functions are considered and their issues of submodularity and monotonicity are addressed in details. All proofs involved in this and the other sections, are given in appendix.

To motivate the first formulation, imagine we are interested in placing sensors to monitor pollution particles in a vast region. Ideally, we would like to place k sensors, which is a given budget, at pollution sources to better predict the particle dynamics for other areas of interest. As such, the placement locations can be obtained by maximizing the directed information from selected location set S to its complement $\overline{S}$ (in the universal set V that contains all candidate sites). Then this type of "causal sensor placement" problems can be written as

$$\underset{S\subseteq V,|S|\leq k}{\operatorname{argmax}} \mathcal{I}(S^n \to \overline{S}^n) \tag{OPT1}$$

Regarding the causal covariate selection problem, the goal is to choose a subset S from a universal set V, such that S has maximal prediction causality to a (or several) target process Y. To leverage sparsity, the cardinality constraints $|S| \leq k$ is also imposed on the number of selected covariates. Again with directed information, this type of subset selection problems reads

$$\underset{S\subseteq V,|S|\leq k}{\operatorname{argmax}} \mathcal{I}(S^n \to Y^n) \tag{OPT2}$$

As a side note, it may seem tempting to use directed information rate as the objective, however we use the accumulative formula here, because it does not require additional regularity and can be used for non-stationary processes. The above two optimizations are hard even in the most simplified cases: Consider a collection of causally independent Gaussian processes, then the above problems are equivalent to the D-optimal design problem, which has been shown to be NP-hard [68]. Unless "P = NP", it is unlikely to find any polynomial algorithm for the maximization, and a resort to tractable approximations is necessary.

2.3.2 Submodularity Analysis

Fortunately, we can show that the objective function of (OPT1), which measures the directed information of selected subsets to unselected ones in V, is submodular.

Theorem 2. *The objective $\mathcal{I}(S^n \to \bar{S}^n)$ as a function of $S \subseteq V$ is submodular.*

The problem is that (OPT1) is not monotonic for all S, which can be seen because $\mathcal{I}(\emptyset \to V)$ and $\mathcal{I}(V \to \emptyset)$ are both equal to 0. However, the deterministic greedy algorithm only has a guaranteed performance when the objective function is monotonic up to $2k$ elements.

In fact, there exist cases such that suboptimal selections in the first few steps would cause a complete failure of the deterministic greedy. Either extra assumptions have to be made, e.g., the objective is monotonically increasing for any $S : |S| \leq 2k$, or we have to reconsider the algorithm to cope with non-monotonicity. The problem of maximizing non-monotonic submodular function has been addressed in literature [69] [70] [71]. In this work we adopt a recent idea in [71], which presents a *randomization* technique to overcome the non-monotonic effect. Compared to other alternatives, it is simple and achieves the best known guarantee.

As for the submodularity of the second objective (OPT2), we make a slight detour and take a look at the property of its "first derivative". For any $x, Y, S \subseteq V$, with $f(S) \triangleq \mathcal{I}(S^n \to Y^n)$ the derivative $f_x(S)$ at S for "direction" x has a more compact form in terms of a causally conditioned directed information,

Proposition 3.

$$f_X(S) = \mathcal{I}(S^n \cup x^n \to Y^n) - \mathcal{I}(S^n \to Y^n) = \mathcal{I}(x^n \to Y^n || S^n) \tag{2.10}$$

Thus the derivative is actually the directed information from the processes x to Y causally conditioned on S. By the first definition of submodularity, if the derivative is decreasing in S, i.e. if $f_x(S) \geq f_x(T)$ for any $S \subseteq T \subseteq V$ and $x \subseteq V \setminus T$, then the objective $\mathcal{I}(S^n \to Y^n)$ is a submodular function. Intuition may suggest this is true since knowing more (conditioning on a larger set) seems to reduce the dependence (and also the causality) of two phenomenon under consideration. However, in general this conjecture is not correct, and a counter example could be constructed by having "explaining away" variables in graphic models. Hence the difficulty encountered for solved (OPT2) is that, in general the objective $\mathcal{I}(S^n \to Y^n)$ is not submodular.

Note that with some extra conditional independence assumptions we can justify its submodularity, as is stated in the following,

Proposition 4. *If for any two processes $s_1, s_2 \in S$, we have the instantaneous conditional independence that $(s_{1t} \perp\!\!\!\perp s_{2t} \mid Y_t)$, then $\mathcal{I}(S^n \to Y^n)$ is a monotonic submodular function of set S.*

In practice the assumption made in the above proposition is hard to check. Yet one may wonder that if the conditional dependence is weak or sparse, possibly existing greedy algorithm still works to some extent, because the submodularity is not seriously deteriorated. This observation suggests that one can define a measure for the degree of submodularity, instead of treating it as a yes-or-no property of set functions. We use this idea to deal with the lack of submodularity. A novel metric, namely Submodularity Index (SmI), is proposed in this work. Notably we will show that, the performance of greedy algorithms is continuously determined by this index. Hence theoretically one can apply greedy heuristics to the maximization of a much broader class of set functions.

2.4 SmI and Performance Bounds

2.4.1 The Submodularity Index and Its Properties

For the ease of notation, we use f to denote a general set function and treat directed information objectives as special realizations. It's worth mentioning that in literature, some effort has already been made to characterize approximate submodularity, such as the ε relaxation of definition (2.7) proposed in [72] for a dictionary selection objective, and the submodular ratio proposed in [64]. Compared to existing works, the SmI suggested in this work (1) is more generally defined for all set functions, (2) does not presume monotonicity, and (3) is more suitable for tasks involving information, influence, and coverage metrics in terms of computational convenience.

To begin with, let's define the *local submodular index* of a function f at location A for candidate set S

$$\varphi_f(S, A) \triangleq \sum_{x \in S} f_x(A) - f_S(A) \tag{2.11}$$

This definition can be considered as an extension of the third definition (2.9) for submodular functions. In essence, it captures the *difference* between the sum of individual effect and aggregated effect on the first derivative of the function.. Moreover, it has the following property:

Proposition 5. *For a given submodular function f, the local submodular index $\varphi_f(S, A)$ is super-modular of S.*

Now we define SmI by minimizing over variables

Definition 2. *For a set function $f : 2^V \to \mathbb{R}$ the submodularity index (SmI) for location set L and cardinality k, denoted by $\lambda_f(L, k)$, is defined as*

$$\lambda_f(L, k) \triangleq \min_{\substack{A \subseteq L \\ S \cap A = \emptyset,\ |S| \leq k}} \varphi_f(S, A) \tag{2.12}$$

Thus SmI is the smallest possible value of local submodularity indexes subject to $|S| \leq k$. Note that we implicitly assume $|S| \geq 2$ in the above definition, as in the cases $|S| = \{0, 1\}$, SmI reduces trivially to 0. Besides, the definition of submodularity can be alternatively posed with SmI:

Lemma 6. *A set function f is submodular if and only if*

$$\lambda_f(L, k) \geq 0 \quad \forall\, L \subseteq V \text{ and } k$$

For functions that are already submodular, SmI measures how strong the submodularity is. We call a function *super-submodular* if its SmI is strictly larger than zeros. On the other hand for functions that are not submodular, SmI provides an indicator on how close the

function is to submodular. We call a function *quasi-submodular* if it has a negative but close to zero SmI.

Direct computation of SmI by solving (2.12) is hard. For the purpose of obtaining a performance guarantee, however, a lower bound of SmI is sufficient and is much easier to compute. Consider the objective of (OPT1), which is already a submodular function. By using proposition 5, we conclude that its local submodular index is a super-modular function for fixed location set. Hence computing (2.12) becomes a cardinality constrained super-modular minimization problem for each location set. Besides, the following decomposition is useful to avoid extra work of directed information estimation:

Proposition 7. *The local submodular index of the function* $\mathcal{I}(\{\bullet\}^n \to \{V \setminus \bullet\}^n)$ *can be decomposed as*

$$\varphi_{\mathcal{I}(\{\bullet\}^n \to \{V\setminus\bullet\}^n)}(S^n, A^n) = \quad \varphi_{H(\{V\setminus\bullet\}^n)}(S^n, A^n) + \sum_{t=1}^{n} \varphi_{H(\{\bullet\}|V^{t-1})}(S_t, A_t)$$

where $H(\bullet)$ *is the entropy function.*

The lower bound of SmI for the objective of (OPT2) is more involved. First observe the following transformation:

Proposition 8.

$$\begin{aligned} &\sum_{x\in S} \mathcal{I}(x^n \to Y^n || A^n) - \mathcal{I}(S^n \to Y^n || A^n) \\ &= \sum_{t=1}^{n} \mathcal{G}_1\left(S^t, \{A^t, Y^{t-1}\}\right) - \sum_{t=1}^{n} \mathcal{G}_1\left(S^t, \{A^t, Y^t\}\right) \end{aligned} \tag{2.13}$$

where the function $\mathcal{G}_k(W, Z) \triangleq \sum_{w\in W} H(w|Z) - kH(W|Z)$ *defined in terms of entropy is super-modular of* W*.*

By further investigating the properties of the function $\mathcal{G}$, we get a lower bound for the SmI of the objective of (OPT2).

Lemma 9. *For any location sets* $L \subseteq V$*, cardinality* k*, and target process set* Y*, we have*

$$\begin{aligned} &\lambda_{\mathcal{I}(\{\bullet\}^n \to Y^n)}(L, k) \\ &\geq \min_{\substack{W\subseteq V \\ |W|\leq |L|+k}} \sum_{t=1}^{n} \left\{ \mathcal{G}_{|L|+k}\left(W^t, Y^{t-1}\right) - \mathcal{G}_{|L|+k}\left(W^t, Y^t\right) \right\} \qquad (2.14) \\ &\geq - \max_{\substack{W\subseteq V \\ |W|\leq |L|+k}} \mathcal{I}(W^n \to Y^n) \geq -\mathcal{I}(V^n \to Y^n) \qquad (2.15) \end{aligned}$$

Since (2.14) is in fact minimizing (maximizing) the difference of two supermodular (submodular) functions, one can use existing approximate or exact algorithms [73] [74] to compute the lower bound. (2.15) is often a weak lower bound, although it is much easier to compute.

2.4.2 Random Greedy and Performance Bound with SmI

With the introduction of SmI, in this subsection we analyze the performance of the random greedy algorithm for maximizing non-monotonic, quasi or super-submodular function in a unified framework. We emphasize this general treatment as it broadens the theoretical guarantee for a much richer class of functions.

The greedy heuristic studied in this work is a randomized variant of the classic greedy algorithm for maximizing cardinality constrained monotonic submodular functions. The idea was recently proposed in [71] [69] to cope with possibly non-monotonic submodular functions. Also in [71], an expected performance bound of $1/e$ was provided. The overall procedure is summarized in algorithm (1) for reference. Note that the random greedy algorithm only requires $O(k|V|)$ calls of function evaluations, making it suitable for large scale problems.

Algorithm 1: Random Greedy for Subset Selection

Input: V, oracle f, cardinality k

$S_0 \leftarrow \phi$;

for $i = 1, ..., k$ **do**

  $M_i = \text{argmax}_{M_i \subseteq V \setminus S_{i-1}, |M_i|=k} \sum_{u \in M_i} f_u(S_i)$;

  Draw u_i uniformly from M_i;

  $S_i \leftarrow S_{i-1} \cup \{u_i\}$;

In order to analyze the performance of the algorithm, we start with two lemmas that reveals more properties of SmI. The following lemma shows that the monotonicity of the first derivative of a general set function f could be controlled by its SmI.

Lemma 10. *Given a set function $f : V \to \mathbb{R}$, and the corresponding SmI $\lambda_f(L, k)$ defined in (2.12), and also let set $B = A \cup \{y_1, ..., y_M\}$ and $x \in \overline{B}$. For an ordering $\{j_1, ..., j_M\}$, define $B_m = A \cup \{y_{j_1}, ..., y_{j_m}\}$, $B_0 = A$, $B_M = B$, we have*

$$f_x(A) - f_x(B) \geq \max_{\{j_1,...,j_M\}} \sum_{m=0}^{M-1} \lambda_f(B_m, 2) \geq M\lambda_f(B, 2) \tag{2.16}$$

Essentially, the above result implies that, for functions lacking strict submodularity, as long as the second order SmI can be lower bounded by some small negative number, the increasing derivative property (hence the submodularity as defined in 2.7) is not seriously degraded. The second lemma provides an SmI dependent bound on the expected value of a function with random arguments.

Lemma 11. *Let the set function $f : V \to \mathbb{R}$ be quasi submodular with $\lambda_f(L, k) \leq 0$. Also let $S(p)$ a random subset of S, with each element appears in $S(p)$ with probability at most p, then*

$$E\left[f(S(p))\right] \geq (1 - p_1)f(\emptyset) + \gamma_{S,p}$$

with $\gamma_{S,p} \triangleq \sum_{i=1}^{|S|}(i-1)p\lambda_f(S_i, 2)$

In the proof of the main theorem, this technical lemma will be used to bound the expected value of the function, as its argument satisfies the probabilistic condition due to the random greedy selection.

Now we present the main theory.

Theorem 12. *For a general (non-monotonic, non-submodular) functions f, let the optimal solution of the cardinality constrained maximization be denoted as S^*, and the solution of the random greedy algorithm be S^g then*

$$E\left[f(S^g)\right] \geq \left(\frac{1}{e} + \frac{\xi^f_{S^g,k}}{E[f\left(S^g\right)]}\right) f(S^*)$$

where $\xi^f_{S^g,k} = \lambda_f(S_g,k) + \frac{k(k-1)}{2}\min\{\lambda_f(S_g,2),0\}$

The role of SmI in determining the performance of the random greedy algorithm is revealed: the bound consist of $1/e \approx 0.3679$ plus a term as a function of SmI. If $SmI = 0$, the $1/e$ bound in previous literature is recovered. For super-submodular functions, as SmI is strictly larger than zero, the theorem provides a stronger guarantee by incorporating SmI. For quasi-submodular functions having negative SmI, although a degraded guarantee is produced, the bound is only slightly deteriorated when SmI is close to zero. In short, the above theorem not only encompasses existing results as special cases, but also suggests that we should view submodularity and monotonicity as a "continuous" property of set functions. Besides, greedy heuristics should not be restricted to the maximization of submodular functions, but can also be applied for "quasi-submodular" functions because a near optimal solution is still achievable theoretically. As such, we can formally define quasi-submodular functions as those having an SmI such that $\frac{\xi^f_{S,k}}{E[f(S)]} > -\frac{1}{e}$.

In the sequel we distinguish two different cases and provide refined bounds when the function is monotonic (not necessarily submodular) or submodular (not necessarily monotonic).

Corollary 1. *For monotonic functions in general, the random greedy algorithm achieves*

$$E\left[f(S^g)\right] \geq \left(1 - \frac{1}{e} + \frac{\lambda'_f(S^g,k)}{E\left[f(S^g)\right]}\right) f(S^*)$$

and deterministic greedy algorithm also achieves

$$f(S^g) \geq \left(1 - \frac{1}{e} + \frac{\lambda'_f(S^g,k)}{f(S^g)}\right) f(S^*)$$

where $\lambda'_f(S^g,k) = \begin{cases} \lambda_f(S^g,k) & \text{if} \quad \lambda_f(S^g,k) < 0 \\ (1-1/e)^2\lambda_f(S^g,k) & \text{if} \quad \lambda_f(S^g,k) \geq 0 \end{cases}$.

Table 2.1: Expected performance guarantee for cardinality constrained submodular maximization with greedy heuristics

Monotonic	Submodular	Classic bound	This work
-	-	NA	$\frac{1}{e} + \frac{\xi^f_{S^g,k}}{E[f(S^g)]}$
-	✓	$\frac{1}{e}$ [71]	$\frac{1}{e} + \frac{\lambda_f(S^g,k)}{E[f(S^g)]}$
✓	-	NA	$1 - \frac{1}{e} + \frac{\lambda_f(S^g,k)}{E[f(S^g)]}$
✓	✓	$1 - \frac{1}{e}$[66]	$1 - \frac{1}{e} + \frac{\gamma\lambda_f(S^g,k)}{E[f(S^g)]}$

We see that in the monotonic case, we get a stronger bound for submodular functions compared to the classic $1 - 1/e \approx 0.6321$ guarantee. Similarly, for quasi submodular functions, the guarantee is degraded but not too much if the negative value of SmD is close to 0. Note that the objective function of (OPT2) fits into this category. For submodular but non-monotonic functions, e.g., the objective function of (OPT1), we have

Corollary 2. *For submodular function that are not necessarily monotonic, the random greedy algorithm has performance*

$$E\left[f(S^g)\right] \geq \left(\frac{1}{e} + \frac{\lambda_f(S^g,k)}{E\left[f(S^g)\right]}\right) f(S^*)$$

From this corollary the role of SmI is made more clear. In table 4.4, we summarize the theoretical guarantees found in this work with SmI, and compare them to classic results.

Another useful observation is that, the performance bound is only related to the ratio $\lambda/f(S_g)$. In fact, in the proof we actually showed stronger results in terms of $\lambda/f(S^*)$ for all cases. Also, a measure of submodularity that is comparable across different set functions would be preferable. These considerations lead us to define the Normalized Submodularity index (NSmI) as

$$\Lambda_f(L,k) \triangleq \frac{\lambda_f(L,k)}{f(L^*)} \tag{2.17}$$

2.5 Causal Graph Structure Learning Algorithms

In this section, we connect the subset selection problems studied in previous section to causal structure learning from a network of time series data. More specifically, it is shown that, assume bounded indegree for each process (time series), the structure learning problem can be reduced to solving (OPT2) for every process in the network. As such, the near optimal random greedy heuristic is applied to establish an efficient algorithm for structure learning.

Furthermore, we discuss directed information estimation from streaming data, and propose a decomposition technique to accelerate the computation.

2.5.1 Causal Structure Learning and its Relation to Causal Subset Selection

A rich body of research exists in literature on the structure learning of graphical models for i.i.d samples, however the problem becomes much more involved when we deal with non-i.i.d dynamic networks of processes. Previously, the structure learning of dynamic networks is usually addressed with multivariate regressive models. For example, in [75], the author proposed an algorithm to identify the topology of network of linear systems. In [76], an alternative is proposed based on Group Lasso. In this work, we adopt the result of a recent work [57], which defined a notion of directed information graph, and proved its equivalence to minimum generative models. First of all, the definition directed information graph is stated as follows,

Definition 3. *[57] A Causal Graph with Directed Information as causality metric, is a directed graph on V with each nodes representing a process, and there is a directed edge from node $i \in V$ to $j \in V$, if and only if*

$$\mathcal{I}(X_i \to X_j || V \setminus \{X_i, X_j\}) > 0 \tag{2.18}$$

Compared to a causal graph based on linear models, a directed information graph is advantageous in that (1) non-linear causality can be captured and Gaussian assumption is not required; (2) the graphical model is equivalent to generative models such as dynamic Bayesian network [57]; (3) confounders can be naturally eliminated due to the causally conditioning in (2.18).

From the above definition, a naïve way of structure learning from data is to check $\mathcal{I}(X_i \to X_j || V \setminus \{X_i, X_j\}) > 0$ for every pair of processes in the network. This $O(|V|^2)$ algorithm seems viable in terms of computational costs, however, to estimate the causally conditioned directed information, i.e., $\mathcal{I}(X_i \to X_j || V \setminus \{X_i, X_j\})$, the joint distribution of all the processes in the network has to be estimated in the first place. This requirement produces serious problems because high dimensional joint distributions are usually hard, if not impossible, to estimate without extra assumptions [77].

The remedy is to realize the following property

Lemma 13. *In a directed information causal graph $\mathbb{G} = (V, \mathcal{E})$, let $\pi(i) \in V$ be the set of all parents of node $i \in V$, then for any other set $W \in V$, we have*

$$\mathcal{I}\left(X_{\pi i} \to X_i\right) \geq \mathcal{I}\left(X_W \to X_i\right) \tag{2.19}$$

This lemma essentially indicates that the complete parents set always has maximal causal influence on its child node (process). Thus, the structure learning problem can be reduced

to solving

$$\underset{S\subseteq V,|S|\leq k}{\operatorname{argmax}}\ \mathcal{I}(S^n \to X_i^n) \tag{2.20}$$

for each node $i \in V$, assuming maximal indegree is k for all nodes. According to Corollary 1, a near optimal approximate solution can be obtained with either random or deterministic greedy search. A deterministic version is summarized in Algorithm 2. Compared to pairwise edge detection, this algorithm only requires estimating a joint distribution of dimension at most $k+1$, which is significantly smaller than $|V|$, the dimension of the full joint distribution.

Algorithm 2: Structure Learning

$\mathbb{G} \leftarrow zeros(N, N)$;
for $i \in V$ **do**
  $(a, \pi_i) \leftarrow \max_{j\in V} \mathcal{I}(X_j^n \to X_i^n)$;
  $d \leftarrow a$, $m \leftarrow 1$;
  while $d \geq \varepsilon$ *&* $m \leq k$ **do**
    $(a', j^*) \leftarrow \max_{j\in V} \mathcal{I}(X_{\pi_i \cup j}^n \to X_i^n)$;
    $d \leftarrow a' - a$, $a \leftarrow a'$, $\pi_i \leftarrow \pi_i \cup j^*$;
    $\mathbb{G}(j^*, i) = 1$, $m \leftarrow m + 1$;

2.5.2 Problem Decomposition and DI Estimation

Let us take another look at (OPT1), which involves solving the problem

$$\underset{S\subseteq V,|S|\leq k}{\operatorname{argmax}}\ \mathcal{I}(S^n \to \overline{S}^n).$$

Although we showed that the objective is submodular and a near optimal solution can be obtained with Algorithm 1, it turns out we still need to estimate directed information from a subset $S \in V$ to its compliments. Again, direct estimation requires the joint distribution of all processes in V, which is problematic when $|V|$ is large. Here the remedy is to realize that directed information graph $\mathbb{G}$ actually provides a sparse representation of the joint distribution. With some algebra, we can find the following decomposition

Lemma 14. *OPT 1 Decomposition*

$$\mathcal{I}(S^n \to \overline{S}^n) = \mathcal{I}\left(\mathcal{C}_{\overline{S}}(S^{n-1}) \to \mathcal{C}_S(\overline{S}^n)\right) + \sum_t I\left(\mathcal{C}_{\overline{S}}(S_t); \mathcal{C}_S(\overline{S}_t) \mid \mathcal{C}_{\overline{S}}(S^{t-1}), \mathcal{C}_S(\overline{S}^{t-1})\right)$$

where $\mathcal{C}_A(B) \triangleq \{X_i \mid X_i \in B, \exists X_j \in A, \mathbb{G}(i,j) = 1\}$ denotes the set of adjacent nodes from A to B. Hence by utilizing the learned structure, the directed information estimation in (OPT1) is reduced to the estimation of local jointly probabilities, which often times have a much smaller dimensionality.

For directed information estimation, in this work we use an estimator recently proposed in [78], in as much as its fast convergence and mild assumptions on the process. Interested

readers may refer to [54][57] and the reference therein for other possibilities. The procedure consists of (1) The estimation of a universal probability assignment, say Q, for the processes under consideration. This is done through the well-known context tree weighting (CTW) algorithm. (2) The estimation of the directed information from process X to Y with

$$\hat{\mathcal{I}}(X^n \to Y^n) \triangleq \hat{H}(Y^n) - \hat{H}(Y^n||X^n) \tag{2.21}$$

where the causal entropy is estimated with

$$\begin{aligned} \hat{H}(Y^n||X^n) &\triangleq -\frac{1}{n}\log Q(Y^n||X^n) \\ Q(Y^n||X^n) &= \prod_{t=1}^{n} Q(Y_t|X^t, Y^{t-1}) \\ \hat{H}(Y^n) &\triangleq \hat{H}(Y^n||\emptyset) \end{aligned} \tag{2.22}$$

Under some technical conditions, it can be shown [78] that the above method converges to the true DI with $O(n^{-1/2}\log n)$ sample complexity, when L_1 norm is used as the distance metric.

2.6 Experiments and Applications

In this section, we conduct experiments to verify the theoretical results on causal subset selection, as well as the proposed causal graph structure learning method. Source code, as well as all data set used in the experiment, can be found at https://github.com/Yuxun/causalsubset.

2.6.1 Data and Setup

The synthesis data is generated with the Bayes network toolbox (BNT) [79] using dynamic Bayesian network models. Two sets of data, D1 and D2, are simulated, each containing 15 and 35 processes, respectively. For simplicity, all processes are $\{0, 1\}$ valued. The processes are created with both simultaneous and historical dependence among each other. In other words, the current value of a particular process X_{it} may depend on current X_{jt}, $j \in V$, and also on historical X_j^{t-1}, $j \in V$. The order (direct memory length) of the historical dependence is set to be 3. The MCMC sampling engine is called to draw 10^4 points for both D1 and D2.

The smart device data set contains per 5mins electricity consumption of 43 plugin devices in Building B90 of LBNL during the period 04/01/2015 to 04/30/2015. Note that data imputation is performed before hand to amend a few missing values, so that all processes can be aligned in time. For the second case study, we collected hourly air pollution (PM2.5) measurement for 36 locations (where a weather station is available) in North California, for the year 2014. Again data preprocessing is conducted to fill missing values. Moreover, we

Table 2.2: Data sets used in experiment

Data set	ID	# process $\|V\|$	sample size n
Synthesis I	D1	15	10000
Synthesis II	D2	35	10000
Smart device	SD	43	8613
Air pollution	PM	36	8752

detrend each time series with a recursive HP-filter [29] to remove long term daily/monthly seasonalities that are not relevant for hourly analyses. The dimensions of the used data sets are summarized in Table 2.2.

For the purpose of directed information estimation, continuous times series, such as those in SD and PM, are normalized and discretized into three levels. The estimation of the universal probability assignment are done through CWT, for which we set the maximal context tree depth to 5. This is sufficient for the synthesis data sets as its memory length is 3, and it is also enough for two real data sets as we are mainly interested in hourly interactions.

2.6.2 Causal Subset Selection Results

Both of the two causal subset selection problems, (OPT1) and (OPT2) are solved with the random greedy algorithm 1 to select set S that has maximal causality. The cardinality constraint is imposed from $k = 2$ to $k = 8$. For comparison purpose, we also conduct an exhaustive search to obtain the true optimal solution and the corresponding objective values. To empirically justify some of the theoretical results, in particular the performance bounds obtained in this work, we compute SmI using the method discussed in Section 2.4.1. Besides, the following two properties to further prune impossible values of SmI:
Assume set B is chosen for computing $\varphi_f(B, S^{g_k})$, let $l = |B|$, $\gamma = \left(1 - \frac{1}{|S^{g_k}|}\right)^l$, then

Proposition 15. *If $\varphi_f(B, S_g) = \xi$, then*

$$f(S^{g_k}) \geq f(S^*) \frac{\gamma}{l + 1 - l\frac{f(S^{g_{k-1}}) + \xi}{f(S^{g_k})}}$$

Proposition 16. *If the increase $\frac{f(S^{g_k} \cup B) - f(S^{g_k})}{f(S^{g_k})} \leq \varepsilon$, then*

$$f(S^{g_k}) \geq f(S^*) \frac{\gamma}{1 + \varepsilon}$$

Table 2.3: Normalized submodularity index (NSmI) for the objectives of (OPT1) and (OPT2) at locations of greedy selections

$k =$	2	3	4	5	6	7	8
SmI (OPT1)	0.382	0.284	0.175	0.082	0.141	0.078	0.074
SmI (OPT2)	-0.305	0.071	-0.068	-0.029	0.030	0.058	0.092

Firstly, (OPT1) on data set D1 is solved. Figure 2.1 shows the results, including an optimal solution by exhaustive search (red-star), random greedy solution (blue-circle), the performance bound in previous work (cyan-triangle), and the bound with SmI (magenta-diamond) in this work. The corresponding normalized SmI values are shown in the first row of Table 2.3. It is seen that the random greedy choice is close to the true optimal choice. In terms of computational time, the greedy method finishes in less than five minutes, however the exhaustive search takes about 10 hours on this small network with $|V| = 15$. Comparing two bounds in Figure 2.1, we see that the theoretical guarantee is greatly improved and a much tighter bound is produced. This is a consequence of the strictly positive SmI values, which makes the guarantee better than $1/e$. The observation justifies that bounds with SmI (Corollary 2) are much better indicators of the performance of the greedy heuristic. On the other hand, there seems to be room for further enhancement of the performance bound, possibly through a more refined computational method for SmI.

Secondly, (OPT2) on data set SD is solved with the monitor of user 1 as the target process Y. The results of random greedy, exhaustive search, and performance bound (Corollary 1) are shown in Figure 2.2, and normalized SmIs are listed in the second row of Table 2.3. Note that the $1 - 1/e$ line (green-triangle) in the figure is only for reference purpose and is NOT a bound of any kind. We observe that although the objective is not submodular, the random greedy algorithm is still near optimal. As we compare the reference line and the bound calculated with SmI (magenta-diamond), we see that the performance guarantee can be either larger or smaller than $1 - 1/e$, depending on the positiveness or negativeness of SmI. By definition SmI measures the submodulairty of a function at some location set. Hence the SmI computed at current greedy selection captures the "local" submodularity of the objective of (OPT2). The main insight gained from this experiment is that, for a function lacking general submodularity, such as the objective function of (OPT2) discussed here, it can be quasi-submodular ($SmI \leq 0$, $SmI \approx 0$) or even super-submodular ($SmI > 0$) at different locations. Accordingly the performance guarantee can be either larger or smaller than $1 - 1/e$, depending on the value of SmI at the current step.

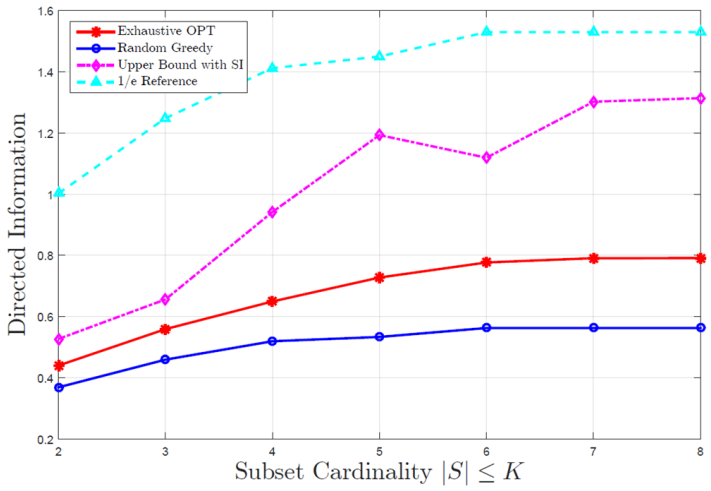


Figure 2.1: Solution and Bounds for (OPT1) on D1

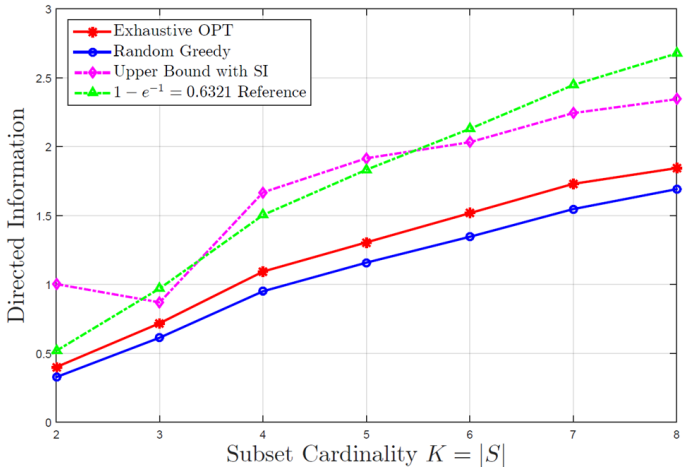


Figure 2.2: Solution and Bounds for (OPT2) on SD

2.6.3 Causal Graph Structure Learning Results

Finally, we test the causal structure learning method proposed in Section 2.5. For Algorithm 2, we set the hyperparameter $\varepsilon = 10^{-3}$ to judge if an increase is achieved, and the maximal in-degree is set to 5. We first use the two synthesis data set D1 and D2 for testing purposes, because their ground truth structures are known and is ready to be compared with. Figure 2.3 and Figure 2.4 demonstrates the results on D1 and D2, respectively. In both two figures, the left subfigure is the ground truth structure, i.e., the dynamic Bayesian networks that are used in the data generation. Note that each node in the figure represent a random process, and an edge from node i to j indicates a causal influence (including both simultaneous and historical). The subfigure on the right shows the causal graph constructed

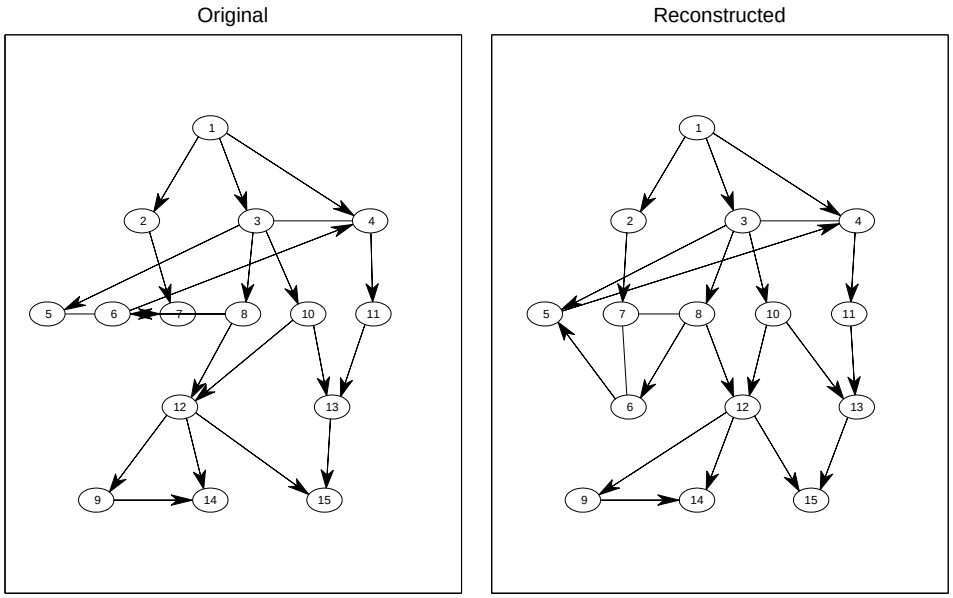


Figure 2.3: Ground truth structure (left) versus Reconstructed causal graph with Algorithm 2 (right), for data set D1

by performing Algorithm 2 on the data sets. Comparing two subfigures in Figure 2.3, we observe that the proposed structure learning method performs almost perfectly. In fact, only the edge $6 \to 4$ is miss detected. On a larger case D2 with $|V| = 35$ processes, the method still works relatively well, correctly reconstructing $41/52$ causal edges. Given that only the maximal indegree (for all nodes) of the causal graph is assumed, these results justify the greedy approximation for the subset selection problem (2.20), as well as the effectiveness of the overall structure learning procedure.

As a more interesting case study, we applied the proposed structure learning method to the PM data set, which contains hourly record of fine particulate matter (PM2.5) for 36 measured locations in north California. The geographic distribution of these locations is shown in the left subfigure of Figure 2.6. And the constructed causal graph is shown in the right subfigure. In this context, the subset selection problem (OPT1) corresponds to selecting "pollution sources". We solve (OPT1) using Algorithm 1, together with the directed information decomposition technique (Lemma 14). Interestingly, we find out that the detected pollution sources are mainly commercial, industrial or transportation centers, such as node 25 (San Francisco) and 7 (Richmond in east bay). Moreover, most of the constructed causal edges are consistent with climatic and geographical implications, such as the edge $29 \to 24$ in the Monteray Bay valley. These results show that the proposed causal structure learning method constitutes a promising tool for data driven sensor placement and source detection.

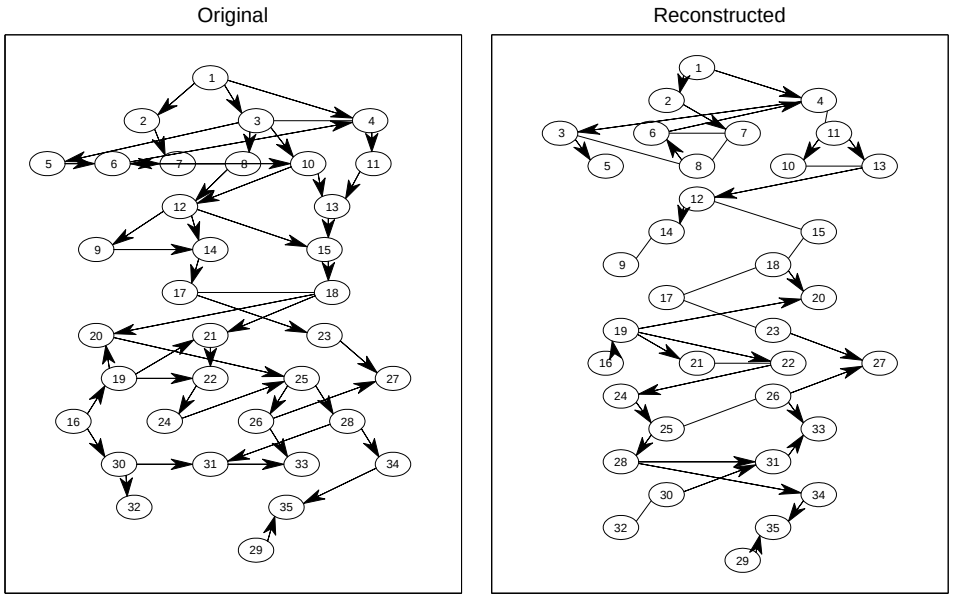


Figure 2.4: Ground truth structure (left) versus Reconstructed causal graph with Algorithm 2 (right), for data set D2

Figure 2.5: 36 measured locations in north California

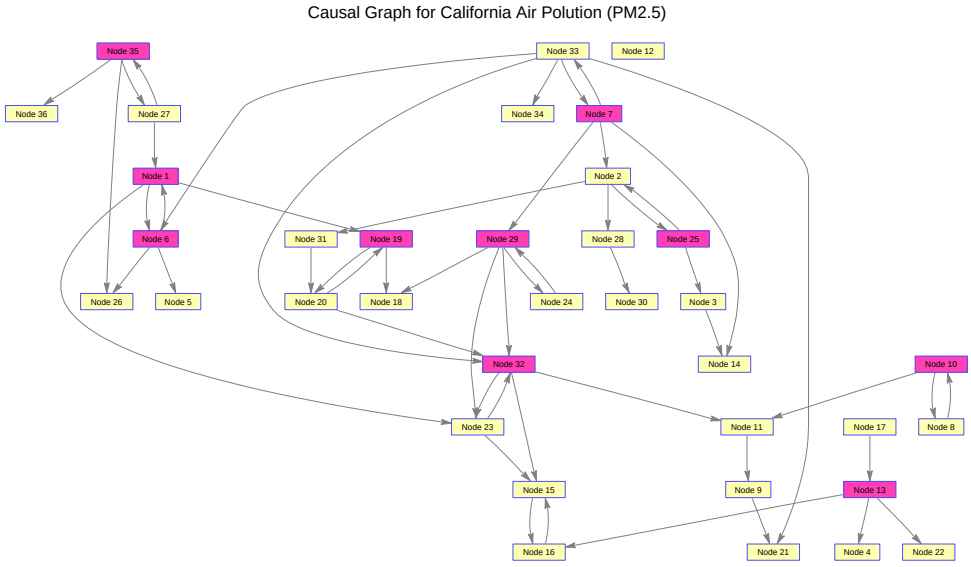


Figure 2.6: Case study: North California air pollution

2.7 Appendix: Proofs

Theorem. *The objective* $\mathcal{I}(A^n \to \bar{A}^n)$ *as a function of* $A \subseteq V$ *is submodular.*

Proof. Let's first show a property of mutual information. At time t, we have

$$\begin{aligned}
&I\left(A^t \cup \{y\}^t; \overline{A \cup \{y\}}_t | \overline{A \cup \{y\}}^{t-1}\right) - I\left(A^t; \overline{A}_t | \overline{A}^{t-1}\right) \\
&= H(V^{t-1}, A_t, y_t) + H\left(\overline{A \cup \{y\}}^t\right) - H(V^t) - H\left(\overline{A \cup \{y\}}^{t-1}\right) \\
&\quad - H(V^{t-1}, A_t) - H(\overline{A}^t) + H(V^t) + H(\overline{A}^{t-1}) \\
&= H(y_t | V^{t-1}, A_t) - H\left(y^t | \overline{A \cup \{y\}}^t\right) + H\left(y^{t-1} | \overline{A \cup \{y\}}^{t-1}\right)
\end{aligned}$$

where we use $I(X, Y|Z) = H(X, Z) + H(Y, Z) - H(X, Y, Z) - H(Z)$. Summing over the last formula over t and canceling telescoping terms, we obtain the following formula by the definition of directed information,

$$\begin{aligned}
&\mathcal{I}\left(A^n \cup \{y^n\} \to \overline{A \cup \{y\}}^n\right) - \mathcal{I}\left(A^n \to \overline{A}^n\right) \\
&= \sum_t H(y_t | V^{t-1}, A_t) - H\left(y^n | \overline{A \cup \{y\}}^n\right) + H\left(y_0\right)
\end{aligned}$$

where we assumed independent initial distribution.[1] Now for any set $B \supseteq A$, by "information

[1] In fact, initial condition does not matter for large t, which is usually true for meaningful DI estimation

never hurt"

$$H(y_t|V^{t-1}, A_t) \geq H(y_t|V^{t-1}, B_t)$$
$$H\left(y^n|\overline{A \cup \{y\}}^n\right) \leq H\left(y^n|\overline{B \cup \{y\}}^n\right)$$

Hence Definition 1 of submodularity is verified. The objective function is submodular. □

Proposition. $f_X(S) = \mathcal{I}(S^n \cup x^n \to Y^n) - \mathcal{I}(S^n \to Y^n) = \mathcal{I}(x^n \to Y^n||S^n)$

Proof. Note the following alternative expression for DI:

$$\begin{aligned}\mathcal{I}(X^n \to Y^n) &= \sum_{t=1}^{n} \left\{H(Y_t|Y^{t-1}) - H(Y_t|Y^{t-1}, X^t)\right\} \\ &= H(Y^n) - \sum_{t=1}^{n} H(Y_t|Y^{t-1}, X^t)\end{aligned} \tag{2.23}$$

and the result can be obtained since $H(Y^n||X^n) \triangleq \sum_{t=1}^{n} H(Y_t|Y^{t-1}, X^t)$, and the directed information from X^n to Y^n when *causally conditioned* on the series Z^n can be written as

$$\mathcal{I}(X^n \to Y^n||Z^n) = H(Y^n||Z^n) - H(Y^n||X^n, Z^n) = \sum_{t=1}^{n} I(X^t; Y_t|Y^{t-1}, Z^t) \tag{2.24}$$

□

Proposition. *If for any two processes $s_1, s_2 \in S$, we have the conditional independence that $(s_{1t} \perp\!\!\!\perp s_{2t} \mid Y_t)$, then $\mathcal{I}(S^n \to Y^n)$ is a monotonic submodular function of set S.*

Proof. In this case, we see that the probabilistic model reduces to "causal naive Bayesian", and the submodulaity follows by check Definition 1 with conditional independence and Proposition 1. □

Lemma. *A set function f is submodular if and only if $\lambda_f(L, k) \geq 0$, $\forall\ L \subseteq V$ and k.*

Proof. Simply take $k = 2$, then $\lambda_{V,2} \geq 0$ implies definition 3 of submodularity, hence f is submodular. For the other direction, assuming f is submodular, then for any $A, S \subseteq V$ and $x_i \in S$ by telescoping

$$\begin{aligned}f(A \cup S) - f(A) &= \sum_{i=1}^{|S|} f(A \cup S_{(i)} \cup x_i) - f(A \cup S_{(i)}) \\ &\leq \sum_{i=1}^{|S|} [f(A \cup x_i) - f(A)]\end{aligned}$$

where $S_{(i)} \triangleq S \setminus \{x_1, ..., x_i\}$ and with the definition of SmI we get $\lambda_{V,k} \geq 0$ □

Lemma. *For any location sets $L \subseteq V$, cardinality k, and target process set Y, we have*

$$\lambda_{\mathcal{I}(\{\bullet\}^n \to Y^n)}(L,k) \geq \min_{\substack{W \subseteq V \\ |W| \leq |L|+k}} \sum_{t=1}^{n} \left\{ \mathcal{G}_{|L|+k}\left(W^t, Y^{t-1}\right) - \mathcal{G}_{|L|+k}\left(W^t, Y^t\right) \right\} \tag{2.25}$$

$$\geq - \max_{\substack{W \subseteq V \\ |W| \leq |L|+k}} \mathcal{I}(W^n \to Y^n) \geq -\mathcal{I}(V^n \to Y^n) \tag{2.26}$$

where the function $\mathcal{G}_k(W,Z) \triangleq \sum_{w \in W} H(w|Z) - kH(W|Z)$ defined in terms of entropy is super-modular of W.

Proof. First note that for any random variable set U, we have

$$\begin{aligned}
&\mathcal{I}(U^n \to Y^n || A^n) \\
&= H(Y^n || A^n) - H(Y^n || U^n, A^n) \\
&= \sum_{t=1}^{n} I(U^t; Y_t | A^t, Y^{t-1}) \\
&= \sum_{t=1}^{n} \left\{ H(U^t | A^t, Y^{t-1}) - H(U^t | A^t, Y^t) \right\}
\end{aligned}$$

Hence by plugging in with x^t, S^t and rearrange, we get

$$\begin{aligned}
\lambda_{Y^n, A^n}(S^n) = \sum_{t=1}^{n} \Bigg\{ & \sum_{x \in S} H(x^t | A^t, Y^{t-1}) - H(S^t | A^t, Y^{t-1}) \\
& - \left[\sum_{x \in S} H(x^t | A^t, Y^t) - H(S^t | A^t, Y^t) \right] \Bigg\} \\
= \sum_{t=1}^{n} & \left\{ \mathcal{G}_{A^t, Y^{t-1}}(S^t) - \mathcal{G}_{A^t, Y^t}(S^t) \right\}
\end{aligned}$$

Let's verify several properties of $\mathcal{G}$
• $\mathcal{G}$ is Supermodular
Remember "information never hurts" inequality, we get

$$\begin{aligned}
&\mathcal{G}_k(W \cup \{y\}, Z) - \mathcal{G}_k(W, Z) = H(y|Z) - kH(y|W, Z) \\
&\leq H(y|Z) - kH(y|L, Z)
\end{aligned}$$

for $W \subseteq L$. Hence by definition $\mathcal{G}_k(W,Z)$ is supermodular.

$$\begin{aligned}
&\mathcal{G}_k(W, Z_1) - \mathcal{G}_k(W, Z_2) \\
&= \sum_{w \in W} \left[H(w|Z_1) - H(w|Z_2) \right] - k \left[H(W|Z_1) - H(W|Z_2) \right]
\end{aligned}$$

is decreasing in k as $H(W|Z_1) \geq H(W|Z_2)$ for $Z_1 \subseteq Z_2$
• $\mathcal{G}$ is Posimodular Recall that a set function is posimodular iif

$$f(S) + f(T) \geq f(S \setminus T) + f(T \setminus S)$$

Let's check

$$\begin{aligned}
&\mathcal{G}_1(S, Z) + \mathcal{G}_1(T, Z) - \mathcal{G}_1(S \setminus T, Z) - \mathcal{G}_1(T \setminus S, Z) \\
&= \sum_{x \in S} H(x|Z) + \sum_{x \in T} H(x|Z) - H(S|Z) - H(T|Z) \\
&\quad - \sum_{x \in S \setminus T} H(x|Z) - \sum_{x \in T \setminus S} H(x|Z) - H(S \setminus T|Z) - H(T \setminus S|Z) \\
&= 2 \sum_{x \in S \cap T} H(x|Z) - H(S \cap T|S \setminus T, Z) - H(S \cap T|T \setminus S, Z) \\
&\geq 2 \sum_{x \in S \cap T} H(x|Z) - 2H(S \cap T|Z) \geq 0
\end{aligned}$$

The last inequality is due to submodularity of $H(\bullet|Z)$ Now let's proof the lemma. Since $H(x|A, Y) = H(x|Y) - H(A|Y) + H(A|x, Y)$, and for any $x \in A$, $H(x|A, Y) = 0$, we have

$$\begin{aligned}
&\mathcal{G}_1\left(S^t, \{A^t, Y^{t-1}\}\right) \\
&= \sum_{x \in S} H(x^t|A^t, Y^{t-1}) - H(S^t|A^t, Y^{t-1}) \\
&= \sum_{x \in S \cup A} H(x^t|A^t, Y^{t-1}) - H(S^t|A^t, Y^{t-1}) \\
&= \sum_{x \in S \cup A} \left\{H(x^t|Y^{t-1}) - H(A^t|Y^{t-1}) + H(A^t|x^t, Y^{t-1})\right\} - H(S^t \cup A^t|Y^{t-1}) + H(A^t|Y^{t-1}) \\
&= \underbrace{\sum_{x \in S \cup A} H(x^t|Y^{t-1}) - H(S^t \cup A^t|Y^{t-1})}_{\mathcal{G}_1(S^t \cup A^t, \{Y^{t-1}\})} \\
&\quad - \sum_{x \in S \cup A} H(A^t|Y^{t-1}) + \sum_{x \in S \cup A} H(A^t|x^t, Y^{t-1}) + H(A^t|Y^{t-1})
\end{aligned}$$

Similar equality could be derived for $\mathcal{G}_1\left(S^t, \{A^t, Y^t\}\right)$, then their difference

$$\begin{aligned}
&\mathcal{G}_1\left(S^t, \{A^t, Y^{t-1}\}\right) - \mathcal{G}_1\left(S^t, \{A^t, Y^t\}\right) \\
&= \mathcal{G}_1\left(S^t \cup A^t, \{Y^{t-1}\}\right) - \mathcal{G}_1\left(S^t \cup A^t, \{Y^t\}\right) + H(A^t|Y^{t-1}) - H(A^t|Y^t) \\
&\quad + \sum_{x \in S \cup A} \left[H(A^t|x^t, Y^{t-1}) - H(A^t|x^t, Y^t)\right] - \sum_{x \in S \cup A} \left[H(A^t|Y^{t-1}) - H(A^t|Y^t)\right]
\end{aligned} \tag{2.27}$$

Now note that $H(A^t|Y^{t-1}) - H(A^t|Y^t) = I(A^t; Y_t|Y^{t-1})$ and $H(A^t|x^t, Y^{t-1}) - H(A^t|x^t Y^t) = I(A^t; Y_t|x^t, Y^{t-1})$ are both positive and increasing in A. We get

$$\begin{aligned}
&- \sum_{x \in S \cup A} \left[H(A^t|Y^{t-1}) - H(A^t|Y^t) \right] \\
\geq &- \sum_{x \in S \cup A} \left[H(A^t \cup x^t|Y^{t-1}) - H(A^t \cup x^t|Y^t) \right] \\
= &- \sum_{x \in S \cup A} \left[H(A^t|x^t, Y^{t-1}) - H(A^t|x^t, Y^t) \right] - \sum_{x \in S \cup A} \left[H(x^t|Y^{t-1}) - H(x^t|Y^t) \right]
\end{aligned}$$

Plug into (2.27) and cancel terms, we get

$$\begin{aligned}
&\mathcal{G}_1\left(S^t, \{A^t, Y^{t-1}\}\right) - \mathcal{G}_1\left(S^t, \{A^t, Y^t\}\right) \\
\geq &- \left[H(A^t \cup S^t|Y^{t-1}) - H(A^t \cup S^t|Y^t) \right] \\
= &- I(A^t \cup S^t; Y_t|Y^{t-1})
\end{aligned}$$

On the other hand, if we relax the third term in (2.27) and use the increasing property of $I(A^t; Y_t|Y^{t-1})$

$$\begin{aligned}
&\mathcal{G}_1\left(S^t, \{A^t, Y^{t-1}\}\right) - \mathcal{G}_1\left(S^t, \{A^t, Y^t\}\right) \\
\geq &\, \mathcal{G}_1\left(S^t \cup A^t, \{Y^{t-1}\}\right) - \mathcal{G}_1\left(S^t \cup A^t, \{Y^t\}\right) \\
&- (|S \cup A| - 1) \left[H(A^t \cup S^t|Y^{t-1}) - H(A^t \cup S^t|Y^t) \right] \\
= &\, \mathcal{G}_{|S \cup A|}\left(S^t \cup A^t, \{Y^{t-1}\}\right) - \mathcal{G}_{|S \cup A|}\left(S^t \cup A^t, \{Y^t\}\right) \\
\geq &\, \mathcal{G}_{|L|+k}\left(S^t \cup A^t, \{Y^{t-1}\}\right) - \mathcal{G}_{|L|+k}\left(S^t \cup A^t, \{Y^t\}\right)
\end{aligned}$$

as $|L| + k \geq |S \cup A|$ and the second properties of function $\mathcal{G}$. Now the inequalities follows from the definition of directed information and the fact that for any $S, A \subseteq V$ that satisfies $A \subseteq L,\ S \cap A = \emptyset,\ |S| \leq k$, they are also feasible solutions for $W = S \cup A :\ |S \cup A| \leq |L| + k$. □

Moreover, in order to avoid additional complexity in estimating entropy terms, he following lemma gives interesting lower and upper bounds in terms of total variation distance:

Lemma. *Let $S = \{x_1, ..., x_k\}$ a set of discrete random variables taking value in finite set $\mathcal{X} : |\mathcal{X}| = d$. Also let Z another random variable and denote the expected conditional total variance difference between $P_{S|Z}$ and $P_{x_1|Z} \otimes P_{x_2|Z} \cdots \otimes P_{x_k|Z}$ as*

$$\delta(S, Z) = \mathbb{E}_Z \left[D_{TV}\left(P_{S|Z} || P_{x_1|Z} \otimes P_{x_2|Z} \cdots \otimes P_{x_k|Z} \right) \right]$$

then

$$\begin{aligned}
2\delta^2(S, Z) \leq &\sum_{x \in S} H(x|Z) - H(S|Z) \\
&\leq \delta(S, Z) log(d^k - 1) + H(\delta(S, Z))
\end{aligned} \tag{2.28}$$

Proof. The lower bound is a direct result from Pinsker's Inequality.

$$\begin{aligned}
&\sum_i H(X_i|Z) - H(S|Z) \\
&= \int_\Omega D_{KL}\left(P_{S|Z}||P_{X_1|Z} \otimes P_{X_2|Z} \cdots \otimes P_{X_k|Z}\right) d\mu(z) \\
&\geq \int_\Omega 2D_{TV}^2\left(P_{S|Z}||P_{X_1|Z} \otimes P_{X_2|Z} \cdots \otimes P_{X_k|Z}\right) d\mu(z) \\
&= 2E_Z[\delta^2]
\end{aligned} \tag{2.29}$$

The upper bound is more involved. First note that for any two random variable $U \sim P_U$ and $U \sim P_V$, further assume that they take value in the same finite discrete set $\mathcal{U}$ and $H(U) \geq H(V)$, then

$$\begin{aligned}
H(U) - H(V) &\leq H(U,V) - H(V) \\
&= H(U|V) \\
&\leq P(e)log(|\mathcal{U}| - 1) + H(e)
\end{aligned} \tag{2.30}$$

the last inequality is due to Fano's inequality, and the error random variable e has distribution $P(e) = P(U \neq V)$. In the sequel, we proceed with the *coupling technique*. In effect, we can couple U and V together such that the coupled joint distribution $\hat{P}(u,v)$ satisfies:

$$\sum_u \hat{P}(u,v) = P_V(v) \ \ \forall v \tag{2.31}$$

$$\sum_v \hat{P}(u,v) = P_U(u) \ \ \forall u \tag{2.32}$$

$$\begin{aligned}
&\sup_{A \subseteq \mathcal{U}} \left\{\hat{P}(U \in A, U \neq V) - \hat{P}(V \in A, U \neq V)\right\} \\
&= \sup_{A \subseteq \mathcal{U}} \hat{P}(U \in A, U \neq V)
\end{aligned} \tag{2.33}$$

In fact, we can construct a jointly probability table for U and V, such that in the table $p(i,j) = 0$ for any $j > i$, and other $p(i,j)$ are subject to our choice, which yields $|\mathcal{U}|(|\mathcal{U}|+1)/2$ variables. The marginal probability $\sum_i p(i,j) = P_V(j)$, $\sum_j p(i,j) = P_U(i)$ impose $2|\mathcal{U}|$ equality constraints on these variables. It is easy to see that only $2|\mathcal{U}| - 1$ constraints are independent (as both rowsum and colsum = 1), hence the linear system has an unique solution when $|\mathcal{U}| = 2$, and infinite number of solutions when $|\mathcal{U}| \geq 3$. In addition, for any $A \subseteq \mathcal{U}$, we have

$$\begin{aligned}
&\hat{P}(U \in A, U \neq V) - \hat{P}(V \in A, U \neq V) \\
&= \hat{P}(U \in A, U \neq V) - \hat{P}(V \in A, U \neq V) + \hat{P}(U \in A, U = V) - \hat{P}(V \in A, U = V) \\
&= \hat{P}(U \in A) - \hat{P}(V \in A)
\end{aligned} \tag{2.34}$$

Hence with the above coupling construction,

$$\begin{aligned}\delta = D_{TV}(P_U||P_V) &= \frac{1}{2}\sum_i |P_U(i) - P_V(i)| \\ &= \sup_{A\subseteq\mathcal{U}} \{P(U\in A) - P(V\in A)\} \\ &= \sup_{A\subseteq\mathcal{U}} \left\{\hat{P}(U\in A) - \hat{P}(V\in A)\right\} \\ &= \sup_{A\subseteq\mathcal{U}} \hat{P}(U\in A, U\neq V) \\ &= \hat{P}(U\neq V)\end{aligned} \tag{2.35}$$

Plug into (2.30), we get $H(U) - H(V) \leq \delta log(|\mathcal{U}| - 1) + H(\delta)$. Now let $P_U = P_{X_1|z} \otimes P_{X_2|z} \cdots \otimes P_{X_k|z}$, and $P_V = P_{S|z}$ we get the desired upper bound. □

The upper bound is sharp as can be verified by the constructive coupling proof, on the other hand, the lower bound could be further improved through similar technique. With this lemma, the SmI of objective OPT2 could be further bounded with total variation distance.

Lemma. *Given a set function $f : V \to \mathbb{R}$, and the corresponding SmI $\lambda_f(L,k)$ defined in (2.12), and also let set $B = A \cup \{y_1, ..., y_M\}$ and $x \in \overline{B}$. For an ordering $\{j_1, ..., j_M\}$, define $B_m = A \cup \{y_{j_1}, ..., y_{j_m}\}$, $B_0 = A$, $B_M = B$, we have*

$$f_x(A) - f_x(B) \geq \max_{\{j_1,...,j_M\}} \sum_{m=0}^{M-1} \lambda_f(B_m, 2) \geq M\lambda_f(B, 2) \tag{2.36}$$

Proof. Let $k = 1$, $S = \{x_1, x_2\}$ and by our definition of SmI

$$\sum_{x\in S} f(A\cup x) - f(A) - [f(A\cup S) - f(A)] \geq \lambda_{A,2}$$

Rearranging gives

$$f(A\cup x_1) - f(A) - [f(A\cup x_1\cup x_2) - f(A\cup x_2)] \geq \lambda_{A,2}$$

or with the notation of derivative

$$f_{x_1}(A) - f_{x_1}(A\cup x_2) \geq \lambda_{A,2} \tag{2.37}$$

This is somewhat a "trimming" property. Now consider $A \subseteq B \subseteq V$. Let's write explicitly $B_j = A \cup \{y_1, ..., y_j\}$, $B_0 = A$, $B_m = B$ with $m = |B| - |A|$, then

$$f_x(B_j) \leq f_x(B_{j-1}) - \lambda_{B_{j-1},2}$$

for $j = 1, ..., m$. Adding the m equations we get

$$f(A) - f(B) \geq \sum_{j=1}^{|B|-|A|} \lambda_{B_j,2} \tag{2.38}$$

Also note that the order of $y_1, ..., y_m$ does not matter. Hence the proposition. □

Lemma. *Let the set function $f : V \to \mathbb{R}$ be quasi submodular with $\lambda_f(L, k) \leq 0$. Also let $S(p)$ a random subset of S, with each element appears in $S(p)$ with probability at most p, then*

$$E\left[f(S(p))\right] \geq (1 - p_1)f(\emptyset) + \gamma_{S,p}$$

with $\gamma_{S,p} \triangleq \sum_{i=1}^{|S|}(i-1)p\lambda_f(S_i, 2)$

Proof. W.l.o.g. assume elements in S are ordered by its probability to be in $S(p)$, i.e. $S = \{u_1, u_2, ..., u_{|S|}\}$ and $p_i = \mathbb{P}(u_i \in S(p)) \geq P(u_j \in S(p)) = p_j$ for any $1 \leq i \leq j \leq |S|$. Define $S_i = \{u_1, u_2, ..., u_i\}$, $S_0 = \emptyset$. Then

$$\begin{aligned}
&E\left[f(S(p))\right] \\
&= E\left[f(\emptyset) + \sum_{i=1}^{|S|} \mathbb{I}_{\{u_i \in S(p)\}} f_{u_i}(S_{i-1} \cap S(p))\right] \\
&\geq E\left[f(\emptyset) + \sum_{i=1}^{|S|} \mathbb{I}_{\{u_i \in S(p)\}} \left[f_{u_i}(S_{i-1}) + (i-1)\lambda_{S_{i-1},2}\right]\right] \\
&= f(\emptyset) + \sum_{i=1}^{|S|} \left[p_i f_{u_i}(S_{i-1}) + (i-1)p_i\lambda_{S_{i-1},2}\right] \\
&= (1-p_1)f(\emptyset) + \sum_{i=1}^{|S|}(p_{i-1} - p_i)f(S_i) + p_{|S|}f(S) + \sum_{i=1}^{|S|}(i-1)p_i\lambda_{S_i,2} \\
&\geq (1-p_1)f(\emptyset) + \sum_{i=1}^{|S|}(i-1)p_i\lambda_{S_i,2} \\
&= (1-p_1)f(\emptyset) + \gamma_{S,p}
\end{aligned}$$

where the first inequality is due to last proposition, and second inequality is a direct result of the assumption that p_i's are in decreasing order. Now if f is strongly submodular, then by the definition of $\lambda_{S,k}$, we see that $\gamma_{S,p} \geq 0$, otherwise if f is only approximately submodular with $\lambda_{S,k} \leq 0$, we have

$$\gamma_{S,p} \geq \sum_{i=1}^{|S|}(i-1)p_1\lambda_{S,2} \geq \frac{|S|(|S|-1)}{2}\lambda_{S,2} \triangleq \beta_S$$

□

Theorem. *For a general (non-monotonic, non-submodular) functions f, let the optimal solution of the cardinality constrained maximization be denoted as S^*, and the solution of random greedy algorithm be S^g then*

$$E\left[f(S^g)\right] \geq \left(\frac{1}{e} + \frac{\xi^f_{S^g,k}}{E[f\left(S^g\right)]}\right) f(S^*)$$

where $\xi^f_{S^g,k} = \lambda_f(S_g, k) + \frac{k(k-1)}{2}\min\{\lambda_f(S_g, 2), 0\}$

Proof. Let $\mathcal{C}^i$ be the event of random choices up to iteration i according to the algorithm. Then by tower property

$$E\left[f_{x_{i+1}}(S_i)\right] = E\left[E\left[f_{x_{i+1}}(S_i)|\mathcal{C}^i\right]\right]$$

Denote S^* the true optimal. The inside expectation is just

$$\begin{aligned} E\left[f_{x_{i+1}}(S_i)|\mathcal{C}^i\right] &= \frac{1}{k}\sum_{x\in M_{i+1}} f_x(S_i) \geq \frac{1}{k}\sum_{x\in S^*\setminus S_i} f_x(S_i) \\ &\geq \frac{1}{k}\left[\lambda_{S_i,|S^*\setminus S_i|} + f(S^*\cup S_i) - f(S_i)\right] \end{aligned} \tag{2.39}$$

in which the first inequality is because M_{i+1} is the maximal, and second inequality is due to the definition of SmI. Now the expectation reads

$$E\left[f_{x_{i+1}}(S_i)\right] \geq \frac{1}{k}\left\{\lambda_{S_i,|S^*\setminus S_i|} + E\left[f(S^*\cup S_i)\right] - E\left[f(S_i)\right]\right\}$$

If f is monotonic, we can further lower bound $f(S^*\cup S_i)$ by $f(S^*)$ and proceed to induction for performance bound, however in the non-monotonic case, this lower bound does not stands any more. In this step the random choice of the algorithm becomes crucial: with lemma lemma:proba, we can show that on average, $f(S^*\cup S_i)$ still has a variant lower bound.

The trick is to notice that with the random greedy algorithm, in each iteration, any element $y \in V\setminus S_i$ will be selected into S_{i+1} with probability at most $1/k$, hence at iteration i, y stays outside of S_i with probability at least $(1-1/k)^i$, or in other words,

$$\mathbb{P}\{y\in S_i\} \leq 1-(1-1/k)^i = p$$

Define function $g(S) = f(S\cup S^*)$, then it is easy to see that g is approximately submodular with $\lambda_{U,n}(g) = \lambda_{U\cup S^*,n}(f)$. Now let's apply the lemma to get

$$\begin{aligned} E\left[f(S^*\cup S_i)\right] = E\left[g(S_i\setminus S^*)\right] &\geq \left(1-\frac{1}{k}\right)^i g(\emptyset) + \beta_{S_i\setminus S^*\cup S^*} \\ &\geq \left(1-\frac{1}{k}\right)^i f(S^*) + \beta_{S_g} \end{aligned}$$

The last inequality is because $S_i \setminus S^* \cup S^* = S_i \subseteq S_g$, and β_S is decreasing in S (as a linear combination of $\lambda_{S,2}$). Continuing with this lower bound on $E\left[f(S^* \cup S_i)\right]$, we get

$$E\left[f_{x_{i+1}}(S_i)\right] \geq \frac{1}{k}\left\{\lambda_{S_g,k} + \beta_{S_g} + \left(1-\frac{1}{k}\right)^i f(S^*) - E\left[f(S_i)\right]\right\}$$

Define $\lambda_{S_g,k} + \beta_{S_g} = -\xi_{S_g}$ a constant with given k, then rearranging yields

$$E\left[f(S_{i+1})\right] - E\left[f(S_i)\right] \geq \frac{1}{k}\left\{\left(1-\frac{1}{k}\right)^i f(S^*) - E\left[f(S_i)\right] - \xi_{S_g}\right\} \tag{2.40}$$

$$E\left[f(S_{i+1})\right] \geq \left(1-\frac{1}{k}\right) E\left[f(S_i)\right] + \frac{1}{k}\left(1-\frac{1}{k}\right)^i f(S^*) - \frac{\xi_{S_g}}{k} \tag{2.41}$$

The last inequality implies that the expected increments made by random greedy algorithm has guarantees, but is deteriorated by the lack of strong submodularity, whose negative effect is incorporated by ξ_{S_g}. Next, we will make use of this inequality with a induction framework and show the overall performance guarantee of the algorithm. Specifically, assume

$$E\left[f(S_i)\right] \geq \frac{i}{k}\left(1-\frac{1}{k}\right)^{i-1} f(S^*) - \frac{\xi_{S_g}}{k}\sum_{j=0}^{i-1}\left(1-\frac{1}{k}\right)^j \tag{2.42}$$

when $i = 1$, we have

$$\begin{aligned} kE\left[f(S_1)\right] &\geq \sum_{x\in S^*} E\left[f(x)\right] \geq E\left[f(S^*)\right] + \lambda_{\emptyset,k} \\ &\geq E\left[f(S^*)\right] + \lambda_{S_g,k} \geq E\left[f(S^*)\right] - \xi_{S_g} \end{aligned}$$

where the first inequality follows because the first step choice S_1 is always maximum, the second and third inequalities are from the SMD definition and its decreasing property, and the last inequality is due to our worst case assumption that f is not submodular and $\beta_{S_g} \leq 0$. Now assume (2.42) is true for any $i' = 1, 2, ...i$, then at $i+1$ step, plugging into (2.41) gives

$$\begin{aligned} &E\left[f(S_{i+1})\right] \\ &\geq \frac{i}{k}\left(1-\frac{1}{k}\right)^i f(S^*) + \frac{1}{k}\left(1-\frac{1}{k}\right)^i f(S^*) - \frac{\xi_{S_g}}{k}\sum_{j=0}^{i}\left(1-\frac{1}{k}\right)^j \\ &= \frac{i+1}{k}\left(1-\frac{1}{k}\right)^i f(S^*) - \frac{\xi_{S_g}}{k}\sum_{j=0}^{i}\left(1-\frac{1}{k}\right)^j \end{aligned}$$

which completes the induction. Let $i = k-1$, we get

$$\begin{aligned} E\left[f(S_g)\right] &\geq \left(1-\frac{1}{k}\right)^{k-1} f(S^*) - \xi_{S_g}\left(1-\left(1-\frac{1}{k}\right)^k\right) \\ &\geq \frac{1}{e} f(S^*) - \xi_{S_g} \geq \left(\frac{1}{e} - \frac{\xi_{S_g}}{E\left[f(S_g)\right]}\right) f(S^*) \end{aligned}$$

□

Proof. COROLLARY 1
This is an easier case, we can start from last line of (2.39) and get

$$\begin{aligned} E\left[f_{x_{i+1}}(S_i)\right] &\geq \frac{1}{k}\left\{\lambda_{S_i,|S^*\setminus S_i|} + E\left[f(S^* \cup S_i)\right] - E\left[f(S_i)\right]\right\} \\ &\geq \frac{1}{k}\left\{\lambda_{S_i,|S^*\setminus S_i|} + E\left[f(S^*)\right] - E\left[f(S_i)\right]\right\} \end{aligned}$$

since f is monotonic, $f(S^* \cup S_i) \geq f(S^*)$. Rearranging yields

$$\begin{aligned} E\left[f(S_{i+1})\right] &\geq \left(1 - \frac{1}{k}\right) E\left[f(S_i)\right] + \frac{1}{k} f(S^*) + \frac{\lambda_{S_i,|S^*\setminus S_i|}}{k} \\ &\geq \left(1 - \frac{1}{k}\right) E\left[f(S_i)\right] + \frac{1}{k} f(S^*) + \frac{\lambda_{S_g,k}}{k} \end{aligned} \tag{2.43}$$

Let's again use induction technique for clarity. Assume

$$E\left[f(S_i)\right] \geq \left[1 - \left(1 - \frac{1}{k}\right)^i\right] f(S^*) + \frac{\lambda_{S_g,k}}{k} \sum_{j=0}^{i-1} \left(1 - \frac{1}{k}\right)^j$$

Then one can easily check that this assumption stands for $i = 1$ with the definition and monotonicity of $\lambda_{U,m}$, and from i to $i+1$ one can just use the induction assumption. Hence we have

$$E\left[f(S_g)\right] \geq \left[1 - \left(1 - \frac{1}{k}\right)^k\right] f(S^*) + \lambda_{S_g,k} \left[1 - \left(1 - \frac{1}{k}\right)^k\right]$$

Now if the function is submodular, we have $\lambda_{S_g,k} \geq 0$, then

$$\begin{aligned} E\left[f(S_g)\right] &\geq \left(1 - \frac{1}{e}\right) f(S^*) + \left(1 - \frac{1}{e}\right) \lambda_{S_g,k} \\ &\geq \left[1 - \frac{1}{e} + \left(1 - \frac{1}{e}\right)^2 \frac{\lambda_{S_g,k}}{E\left[f(S_g)\right]}\right] f(S^*) \end{aligned}$$

where we have used $E\left[f(S_g)\right] \geq \left(1 - \frac{1}{e}\right) f(S^*)$ in the second inequality. On the other hand, if $\lambda_{S_g,k} \leq 0$, we get

$$\begin{aligned} E\left[f(S_g)\right] &\geq \left(1 - \frac{1}{e}\right) f(S^*) + \lambda_{S_g,k} \\ &\geq \left(1 - \frac{1}{e} + \frac{\lambda_{S_g,k}}{E\left[f(S_g)\right]}\right) f(S^*) \end{aligned}$$

□

Proof. Corollary 2
Simply note that in this case Lemma lemma:proba becomes $E\left[f(S(p))\right] \geq (1-p_1)f(\emptyset)$, and we just follow the lines of proof of Theorem 3 with ξ_{S_g} replaced by $\lambda_{S_g,k}$. □

Lemma. *OPT 1 Decomposition*

$$\mathcal{I}(S^n \to \overline{S}^n) = \mathcal{I}\left(\mathcal{C}_{\overline{S}}(S^{n-1}) \to \mathcal{C}_S(\overline{S}^n)\right) + \sum_t I\left(\mathcal{C}_{\overline{S}}(S_t); \mathcal{C}_S(\overline{S}_t) \mid \mathcal{C}_{\overline{S}}(S^{t-1}), \mathcal{C}_S(\overline{S}^{t-1})\right)$$

Proof. Proof of DI decomposition

$$\begin{aligned}
&\mathcal{I}(S^n \to \overline{S}^n) \\
&= \mathcal{I}\left(\mathcal{C}_{\overline{S}}(S^n) \cup \mathcal{N}_{\overline{S}}(S^n) \to \mathcal{C}_S(\overline{S}^n) \cup \mathcal{N}_S(\overline{S}^n)\right) \\
&= \mathcal{I}\left(\mathcal{C}_{\overline{S}}(S^n) \to \mathcal{C}_S(\overline{S}^n) \cup \mathcal{N}_S(\overline{S}^n)\right) + \mathcal{I}\left(\mathcal{N}_{\overline{S}}(S^n) \to \mathcal{C}_S(\overline{S}^n) \cup \mathcal{N}_S(\overline{S}^n) \mid\mid \mathcal{C}_{\overline{S}}(S^n)\right) \\
&= \mathcal{I}\left(\mathcal{C}_{\overline{S}}(S^n) \to \mathcal{C}_S(\overline{S}^n) \cup \mathcal{N}_S(\overline{S}^n)\right) \\
&= \mathcal{I}\left(\mathcal{C}_{\overline{S}}(S^n) \to \mathcal{N}_S(\overline{S}^n) \mid\mid \mathcal{C}_S(\overline{S}^n)\right) + \mathcal{I}\left(\mathcal{C}_{\overline{S}}(S^n) \to \mathcal{C}_S(\overline{S}^n) \mid\mid \mathcal{N}_S(\overline{S}^{n-1})\right) \\
&= \mathcal{I}\left(\mathcal{C}_{\overline{S}}(S^n) \to \mathcal{C}_S(\overline{S}^n) \mid\mid \mathcal{N}_S(\overline{S}^{n-1})\right) \\
&= \sum_t I\left(\mathcal{C}_{\overline{S}}(S^t); \mathcal{C}_S(\overline{S}_t) \mid \mathcal{C}_S(\overline{S}^{t-1}), \mathcal{N}_S(\overline{S}^{t-1})\right) \\
&= \sum_t I\left(\mathcal{C}_{\overline{S}}(S^{t-1}); \mathcal{C}_S(\overline{S}_t) \mid \mathcal{C}_S(\overline{S}^{t-1}), \mathcal{N}_S(\overline{S}^{t-1})\right) \\
&\qquad + \sum_t I\left(\mathcal{C}_{\overline{S}}(S_t); \mathcal{C}_S(\overline{S}_t) \mid \mathcal{C}_{\overline{S}}(S^{t-1}), \mathcal{C}_S(\overline{S}^{t-1}), \mathcal{N}_S(\overline{S}^{t-1})\right) \\
&= \sum_t I\left(\mathcal{C}_{\overline{S}}(S^{t-1}); \mathcal{C}_S(\overline{S}_t) \mid \mathcal{C}_S(\overline{S}^{t-1})\right) + \sum_t I\left(\mathcal{C}_{\overline{S}}(S_t); \mathcal{C}_S(\overline{S}_t) \mid \mathcal{C}_{\overline{S}}(S^{t-1}), \mathcal{C}_S(\overline{S}^{t-1})\right) \\
&= \mathcal{I}\left(\mathcal{C}_{\overline{S}}(S^{n-1}) \to \mathcal{C}_S(\overline{S}^n)\right) + \sum_t I\left(\mathcal{C}_{\overline{S}}(S_t); \mathcal{C}_S(\overline{S}_t) \mid \mathcal{C}_{\overline{S}}(S^{t-1}), \mathcal{C}_S(\overline{S}^{t-1})\right)
\end{aligned}$$

□

Chapter 3

Learning Outliers and Novelty from Multiple Time Series

Data sets collected from modern cyber physical systems are mostly real-time measurements of system behaviors or characteristics. For example, recent advances in sensor networks and information technologies in smart buildings have given researchers access to large amounts of time series data, including but not limited to environmental measurements (temperature, humidity, air quality), energy consumption related records (HVAC operation, lighting, plug loads), and occupant data (individual behavior, presence, location), etc. In this chapter, two closely related tasks, namely outlier and novelty detection, are considered. More specifically, we discuss both parametric and non-parametric methods that integrate the interactions among multiple correlated time series. Note that the interaction structure is assumed to be known.

3.1 Introduction: Outlier and Novelty Detection in Multiple Time Series

General outlier detection is a broad topic that is usually studied separately in the context of particular domain application. From a statistical learning perspective, however, outlier detection techniques can be categorized according to their input data types, including but are not limited to independent and identically distributed observations [80], high-dimensional data [81], time series [82], structural data such as graphs and network [83, 84], etc. A detailed exposition of general outlier detection techniques is beyond the scope of this chapter. The readers are referred to [85, 86, 87] and the references therein for an extensive overview.

This section is focused on the *outlier detection in multiple correlated times series*, which takes the sensor network measurements in modern cyber-physical systems as input, and aims at detecting abnormal system states, unusual behaviors, abrupt changes, etc., for preventive operation and system diagnosis. For example, to automate the fault detection and diagnosis (FDD) procedure in smart buildings, sensor networks are deployed to monitor the state

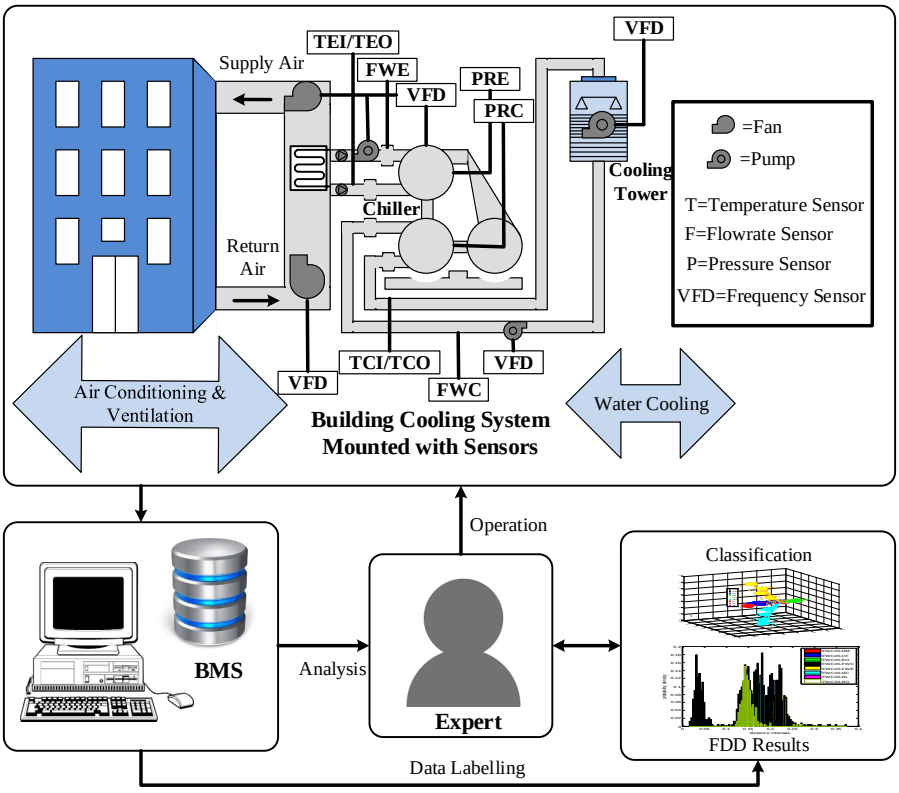


Figure 3.1: Example 1: Building FDD system, including deployed sensor network, data base, and outlier detection algorithm.

of critical components of the building, such as environmental variables, HVAC operation, lighting, plug load, etc., and an outlier detection algorithm is implemented to process the observed data for FDD. Such a decision support system, realized in one of our joint works [88], is depicted in Fig 3.1. Another example involves the fault detection in power distribution networks. The recent advancements in the high fidelity sensing technology, in particular micro-phasor measurement units (μPMUs), enable operators to detect system dynamics that would otherwise be unobservable in distribution networks. The data acquired from μPMUs is essentially multiple correlated voltage and current readings that are used as the input to an outlier detection algorithm. The overall cyber-physical system configuration, based on one of our previous work [89, 90], is illustrated in Fig 3.2. Besides traditional application such as anomaly discovery, methods of outlier detection can also be used to reveal interesting behavior related patterns in CPS. With the ubiquity of WiFi infrastructure and WiFi enabled mobile devices, WiFi has become the primary sensing techniques for occupancy sensing in indoor environment [91, 92, 93, 94, 95]. An occupancy adaptive lighting control system is proposed in [96, 35]. Moreover, a novel device-free occupancy sensing platform is developed

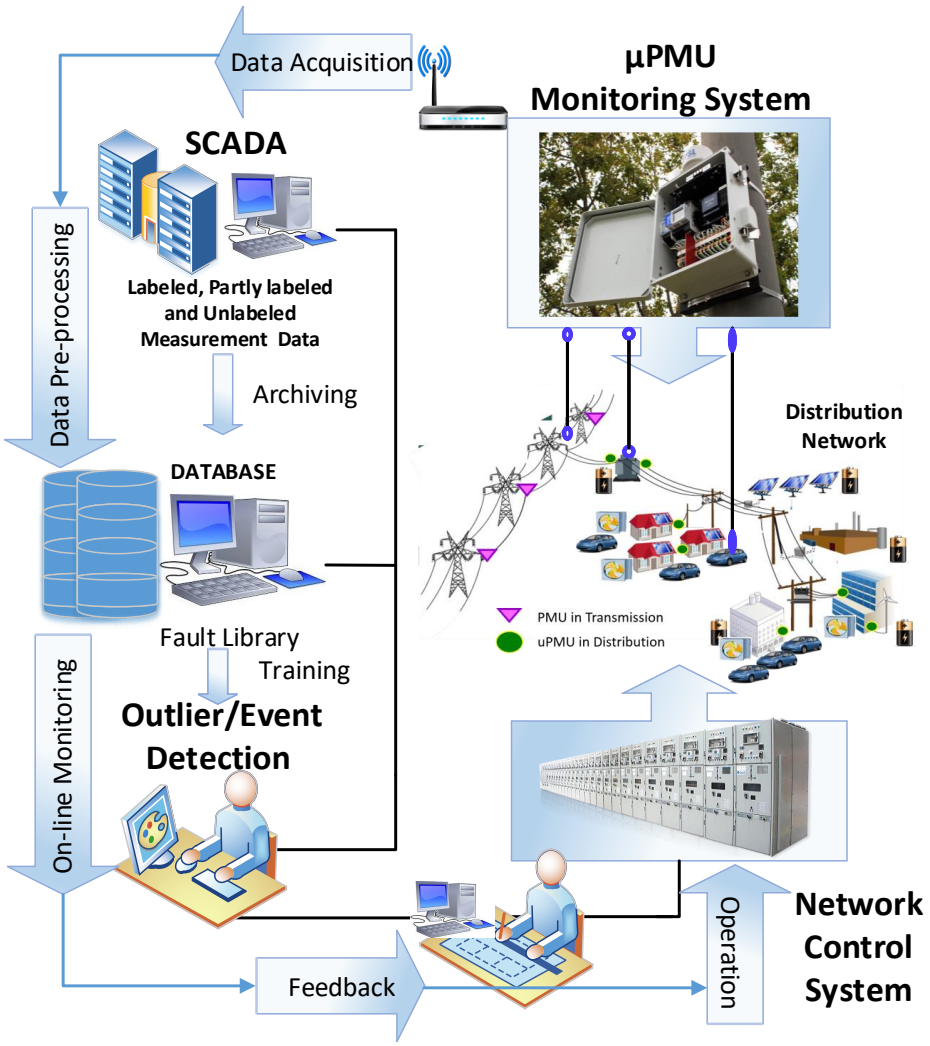


Figure 3.2: Example 2: Fault detection in power distribution networks, with μPMU and detection algorithm deployed.

to provide fine-grained occupancy information, such as occupancy detection [97] and crowd counting [98]. In the aforementioned works, the core occupancy sensing algorithm can be realized by performing outlier detection with WiFi signals. Fig 3.3 shows the configuration of WiFi routers, the collected radio frequency signals, and the outlier detection based decision support system.

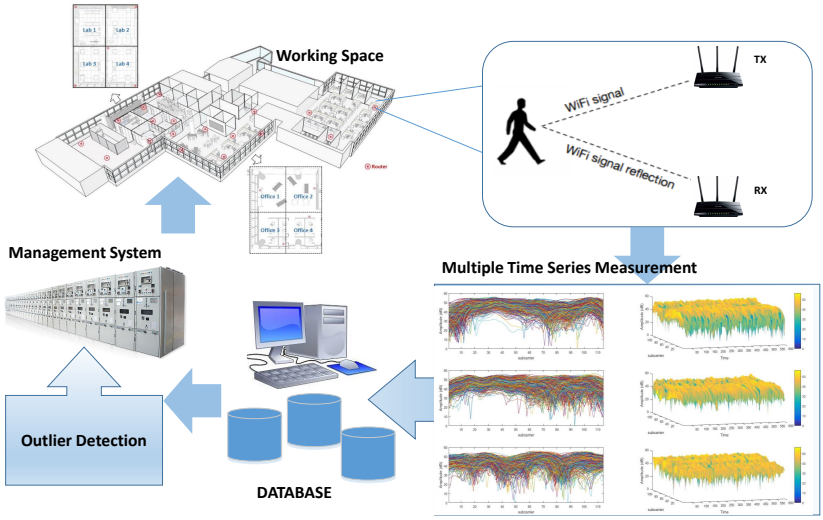


Figure 3.3: Example 3: Non-intrusive occupancy detection with WiFi signal.

Depending on different views of the data generating process, methods for outlier detection in time series can be categorized into the following categories:

- Physical model based methods. The underlying assumption is that the observed time series data is generated from a known dynamic system. As such, the problem is reduced to comparing the system behavior, estimated through measurement and dynamic equations, to the expected behavior when the system is in a certain state (normal or abnormal)[99]. To list a few, model based applications range from cyber attacks identification in power system [100, 101], fault diagnosis for switching converters [102], and fault-detection of engine systems [103]. Recently, model based approaches have also been used in combination with time series analysis to establish semi model based algorithms [104, 105]. This type of approaches rely heavily on correctness of the dynamical model of the system, as well as system analytic tools such as real time state estimators, parameter estimation, parity equations etc. Their limitations are obvious since: (1) dynamics of a system may be hard to specify in many cases and they have non-linear structures, and (2) more and more applications are dealing with complex system with randomness, the high dimensionality and inherent uncertainty significantly deteriorate the reliability and accuracy of dynamic models.

- Signal processing based filtering methods. Those approaches implicitly assumes that the "normal" component of the time series has sparse representation in the frequency or wavelet domain. Hence the outlier detection problem is reduced to spectral analysis using low pass or band pass filters [106], or denoising/signal reconstruction using spectral or wavelet techniques [107]. It is worth pointing out that the signal processing based methods have close ties with the regularized basis function expansion method

in statistical learning. For example, the adaptive wavelet denoising method known as SURE shrinkage [108] is essentially the L_1 regularized wavelet basis expansion.

- Statistical learning based method. The key is to model the characteristics of the normal state, e.g., the support of its distribution, its sparse representation, or its smooth component, with parametric or non-parametric learning tools. As a large amount of data is made available by the advancements in sensor network and information technology, this approach is receiving increasing attention in both application and research domains. Ignoring the temporal dependence, many classic machine learning tools, such as the Kernel Principle Component Analysis (kPCA), Partial Least Squares (PLS), one class SVM, etc., have been widely applied to various fields. When the temporal dependece is informative, miscellaneous time series modeling and analysis tools, ranging from simple linear regression to complicated AMRIA models and from parametric dynamic Bayeisan networks to non-parametric regression methods, can be adopted. Readers are referred to [109, 85] and the references therein for a comprehensive survey.

However, few works have addressed the outlier detection problem for multiple correlated time series. In this section, we propose two learning frameworks, one based on non-parametric smoothing and the other based on the collaborative filtering of HMMs, to learn trends and identify outliers/novelty from a rich family of multiple time series. For each of the learning formulation, we propose efficient optimization algorithms and test them on real-world data set generated from CPS.

Before proceeding to any technical details, we standardize our notation by using a matrix $X^{M\times T}$ to represent all time series measurements for T time steps and M streams. Note that for sensor network applications we usually have $M = K \times L$ where K is the number of channels of each sensor and L the number of sensors installed in the network. To represent the dependence among streams, a "contextual" matrix $C^{M\times M}$ is designated to store the pair-wise correlations. Also for the ease of discussion, we adopt the notion of Network of Time Series (NoT):

Definition 4. *A Network of Time Series (NoT) is defined as the triplet $\mathcal{G} = \{X, C, d\}$, where $X \in \mathbb{R}^{M\times T}$ is a collection of M time series of T time steps, $C \in \mathbb{R}^{M\times M}$ is the contextual matrix and d a dictionary that maps each dimension or stream of X to an entry in C.*

Regarding outlier detection, we adopt the convention that the abnormality or novelty of a observation X_{it} is defined as the deviation between estimated (expected) value $\widehat{X}_{it}$ and real measurement X_{it}. Hence the problem of novelty detection reads,

Problem 1. *Given $\mathcal{G} = \{X, C, d\}$, estimate $\widehat{X}_{it}$, $\forall i, t$. Then compute $l\left(X_{it}, \widehat{X}_{it}\right)$ as the index of novelty, where $l(\cdot,\cdot)$ is a metric function $\mathbb{R}\times\mathbb{R} \mapsto \mathbb{R}$.*

Hence the core of the novelty detection problem is an estimation problem, for which both temporal dependence and inter-series correlation should be taken into account.

3.2 A Simple Nonparametric Approach

The first method we propose borrows ideas from two separate yet closely related research domains, i.e., time series de-trending in economics and non-parametric regression in statistical learning. The learning formulation for multiple time series data is quite intuitive, and it is possible to extend this approach to other data types with the introduction of the Bregman Divergence. To solve the learning problem a simple yet efficient coordinate descent algorithm is proposed. Source code and experimental data can be found at https://github.com/Yuxun/nonParamDetection.

3.2.1 Problem Formulation

We start by considering the following decomposition for a single time series x_t:

$$x_t = u_t + w_t \qquad \forall t \tag{3.1}$$

where the new time series u_t represents the trend component in the terminology of economics, and the second term w_t contains the so called cyclical component and noises of the original time series [110][1]. As such, outlier or novelty can be intuitively defined as elements that deviate significantly from the general trend. In order to find the trend component, one can simply optimize over a "fitness" and "smoothness" trade-off:

$$\min_{u_0,\cdots,u_T} \sum_{t=1}^{T} l(x_t, u_t) + \lambda \Omega(u_0, \cdots, u_T) \tag{3.2}$$

where $l(\cdot,\cdot)$ and $\Omega(\cdot)$ are loss functions imposed on "fitness" and "smoothness", respectively. The above formulation is also closely related with the non-parametric regression method in statistical learning [111], in which a regression function is found by minimizing the L_2 loss with second derivative regularization. Similarly, when dealing with time series data containing discrete-time, continuous-value records, one can substantiate objective (3.2) as follows:

$$\min_{u_0,\cdots,u_T} \sum_{t=1}^{T} (x_t - u_t)^2 + \lambda \sum_{t=1}^{T} (\nabla_t^2 u_t)^2 \tag{3.3}$$

where ∇_t^2 is the second order difference operator defined by:

$$\nabla_t^2 u_t = \begin{cases} 0 & t = 1 \\ u_{t+1} + u_{t-1} - 2u_t & 2 \le t \le T-1 \\ 0 & t = T \end{cases} \tag{3.4}$$

Like the second order derivative regularization used in non-parametric regression, the above aggregated second order differences also measures the smoothness of the entire sequence.

[1] Hence one can decompose this term into $w_t = c_t + \varepsilon_t$ for further analysis

By solving the convex quadratic optimization problem (3.3), one is able to find the trend component u_t. Any data point that significantly deviate from the trend is an outlier or novelty point. The weighting parameter λ is called the smoothness parameter, which should be tuned according to the application purpose using model selection techniques. This will be detailed in the experiment section of this chapter. It is worth pointing out that the solution to (3.3) is called the Hodrick-Prescott filter in economic time series analysis [112].

Now we extend the above non-parametric framework to handle multiple time series that are correlated with each other. Notation-wise, given multiple time series data $X \in \mathbb{R}^{M\times T}$, we denote the t^{th} element of the m^{th} time series by x_{mt}, i.e., x_{mt} is the $(m,t)^{th}$ entry of the data matrix X. Also, the boldface $\boldsymbol{x}_m$ is used to represent the row vector $[x_{m1}, \cdots, x_{mT}]$. Similarly, $\boldsymbol{u}_m = [u_{m1}, \cdots, u_{mT}]$. Now consider minimizing the following objective:

$$\min_{\boldsymbol{u}_1,\cdots,\boldsymbol{u}_M} \sum_{m=1}^{M}\sum_{t=1}^{T}(x_{mt}-u_{mt})^2 + \lambda_1 \sum_{m=1}^{M}\sum_{t=2}^{T-2}\left(\nabla_t^2 u_{mt}\right)^2 + \lambda_2 \sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M}\sum_{t=2}^{T-1}\left[\nabla_t^2(u_{it}-C_{ij}u_{jt})\right]^2 \tag{3.5}$$

where λ_1 and λ_2 are two regularization hyper-parameters, and C is the standardized covariance matrix with entries

$$C_{ij} = \operatorname{cov}(\boldsymbol{x}_i, \boldsymbol{x}_j)\left(\operatorname{var}(\boldsymbol{x}_j)\right)^{-1} \tag{3.6}$$

The intuition for the first two terms in (3.5) is straightforward: we simply aggregate the fitness and smoothness objectives of M times sequences. The motivation for the third term is the following: Since the linear least square estimator (LLSE) [113] of $\boldsymbol{u}_i$ given $\boldsymbol{u}_j$ reads

$$\mathbb{E}[\boldsymbol{u}_i] - \operatorname{cov}(\boldsymbol{u}_i, \boldsymbol{u}_j)\left(\operatorname{var}(\boldsymbol{u}_j)\right)^{-1}(\boldsymbol{u}_j - \mathbb{E}[\boldsymbol{u}_j]).$$

In the case where the two trends are ideally correlated, $\boldsymbol{u}_i - \operatorname{cov}(\boldsymbol{u}_i, \boldsymbol{u}_j)\left(\operatorname{var}(\boldsymbol{u}_j)\right)^{-1}\boldsymbol{u}_j$ should be a constant sequence. Consider estimating the covariance of U by that of the noisy X, and relax the harsh "constant" requirement to smoothness, then with the same usage of second order difference, the third term imposes the smoothness of the sequence $\boldsymbol{u}_i - C_{ij}\boldsymbol{u}_j$, which is aggregated over all pairwise combinations. The objective (3.5) constitutes a non-parametric learning formulation for multiple, interacted time series. The optimization problem is still convex quadratic, and in a tensor form can be written as:

$$\min_{U\in\mathbb{R}^{M\times T}} \|X-U\|_{\mathcal{F}}^2 + \lambda_1 \operatorname{tr}\left(UQ^TQU^T\right) + \lambda_2 \operatorname{tr}\left([U\otimes \boldsymbol{e} - W(\boldsymbol{e}\otimes U)]Q^TQ[U\otimes \boldsymbol{e} - W(\boldsymbol{e}\otimes U)]^T\right) \tag{3.7}$$

in which $\|\cdot\|_{\mathcal{F}}$ is the Frobenius norm or the $L_{2,2}$ norm of a matrix, $\operatorname{tr}(\cdot)$ computes the trace of a squared matrix, and $\otimes$ is the tensor product [114]. The vector $\boldsymbol{e}$ has dimension M and contains all ones, i.e.,

$$\boldsymbol{e} = \underbrace{[1, 1, \cdots, 1]}_{M1s}^T \tag{3.8}$$

The matrix Q performs second order difference operation and can be specified as

$$Q = \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 & -2 & 1 \end{bmatrix}_{(T-2)\times T} \tag{3.9}$$

The matrix W has dimension $M^2 \times M^2$, and encodes the computation of pairwise residuals. More specifically, W is defined block-wise by

$$W = \begin{bmatrix} H_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & H_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{H_M} \end{bmatrix}_{M^2\times M^2}, \qquad H_m = \operatorname{diag}\left([C_{m1}, C_{m2}, \cdots, C_{mM}]\right) \tag{3.10}$$

Noting the similarity of the regularization with various versions of multi-task learning, the proposed method learning formulation can be viewed as multi-task extension of time series trend identification.

3.2.2 Extension to the Exponential Family

The previous section is focused on time series having continuous values. Many CPS measurements, however, may be non-negative or categorical depending on the data generating process. For example, the count of the number of occupants in a building should be modeled after a Poisson distribution instead of being treated as a continuous real value. Given that consideration, this section is devoted to extend the smoothing method developed in last section to time series with exponential family marginal distributions. The learning formulation still has a form similar to (3.5), which optimizes the trade-off between fitness and both temporal smoothness of each time series and inter-series smoothness.

It is helpful to recall some definitions to begin with:

Definition 5. *The **Bregman Divergence** of any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, with respect to some arbitrary differentiable strictly convex function $F : \mathbb{R}^n \to \mathbb{R}$ is defined by*

$$B_F(\boldsymbol{x}, \boldsymbol{y}) = F(\boldsymbol{x}) - F(\boldsymbol{y}) - (\boldsymbol{x} - \boldsymbol{y}) \cdot F'(\boldsymbol{y}) \tag{3.11}$$

One can think of the Bregman Divergence as simply the nonlinear tail of the Taylor expansion of $F(\boldsymbol{x})$ around $\boldsymbol{y}$. Note that the Bregman Divergence is not symmetric, however, it holds that $B_F(\boldsymbol{x}, \boldsymbol{y}) = 0$ iff $\boldsymbol{x} = \boldsymbol{y}$.

Definition 6. *A family of distributions is said to belong to* ***Exponential Family*** *in canonical form if the probability density function, or probability mass function for discrete distributions, can be written as*

$$f_X(\boldsymbol{x}|\boldsymbol{\theta}) = h(\boldsymbol{x}) \exp\{\boldsymbol{\theta} \cdot T(\boldsymbol{x}) - A(\boldsymbol{\theta})\} \tag{3.12}$$

where the parameter vector $\boldsymbol{\theta}$ is called the natural parameter of the distribution, and $T()$ the sufficient statistic.

The normalization factor

$$A(\boldsymbol{\theta}) = \log \int h(\boldsymbol{x}) \exp\{\boldsymbol{\theta} \cdot T(\boldsymbol{x})\} \, d\boldsymbol{x} \tag{3.13}$$

is strictly convex and plays an important role in characterizing members of the exponential family. In particular one can show that the cumulant generating function is

$$K(\lambda) = A(\lambda + \boldsymbol{\theta}) - A(\boldsymbol{\theta})$$

with which one can obtain that,

$$E_{\boldsymbol{\theta}}[T(\boldsymbol{x})] = \nabla_{\boldsymbol{\theta}} A(\boldsymbol{\theta}) \triangleq a(\boldsymbol{\theta}) \tag{3.14}$$

When Bregman Divergence is used in measuring the fitness of observed data to a parametrized exponential family distribution, the following property shows that Bregman divergence is directly related with log-likelihood:

Proposition 17. *Define a dual function associated with the exponential family*

$$F(a(\boldsymbol{\theta})) \triangleq \boldsymbol{\theta} \cdot a(\boldsymbol{\theta}) - A(\boldsymbol{\theta}) \tag{3.15}$$

then $F(\mu)$ is strictly convex in $\mu = a(\boldsymbol{\theta})$. In addition,

$$B_F(T(\boldsymbol{x})||a(\boldsymbol{\theta})) \propto -logP(T(\boldsymbol{x})|\boldsymbol{\theta}) \propto A(\boldsymbol{\theta}) - T(\boldsymbol{x}) \cdot \boldsymbol{\theta} \tag{3.16}$$

Proof. Treating $\boldsymbol{\theta}$ as a function of μ, and taking derivative of $F(a(\boldsymbol{\theta}))$ with respect to μ, we get

$$\nabla_\mu F(\mu) = f(\mu) = \boldsymbol{\theta} + \frac{\partial \boldsymbol{\theta}}{\partial \mu}\mu - \frac{\partial \boldsymbol{\theta}}{\partial \mu}\mu = \boldsymbol{\theta} \tag{3.17}$$

which is in effect the inverse of $a(\boldsymbol{\theta})$, i.e.,$\nabla_\mu F(\mu) = \boldsymbol{\theta} = a^{-1}(\mu)$. From the strict convexity of $A(\boldsymbol{\theta})$, it is guaranteed that this inverse always exists. Moreover, since $a(\boldsymbol{\theta})$ has a positive definite Jacobian, its inverse $a^{-1}(\mu)$ also has a positive definite jacobian. Hence $F(\mu)$ is

strictly convex. In real analysis, $F(\mu)$ is also called the dual convex function of $A(\boldsymbol{\theta})$ [115]. Using this function in the Bregman divergence for $\boldsymbol{x}$ and $a(\boldsymbol{\theta})$, we get

$$\begin{aligned} B_F(T(\boldsymbol{x})||a(\boldsymbol{\theta})) &= F(T(\boldsymbol{x})) - F((\boldsymbol{\theta})) - (T(\boldsymbol{x}) - a(\boldsymbol{\theta})) \cdot \nabla F(a(\boldsymbol{\theta})) \\ &= F(T(\boldsymbol{x})) - a(\boldsymbol{\theta}) + A(\boldsymbol{\theta}) - (T(\boldsymbol{x}) - a(\boldsymbol{\theta})) \cdot \boldsymbol{\theta} \\ &= F(T(\boldsymbol{x})) + A(\boldsymbol{\theta}) - T(\boldsymbol{x}) \cdot \boldsymbol{\theta} \end{aligned}$$

On the other hand, since the log likelihood of the exponential family is just

$$\log P(T(\boldsymbol{x})|\boldsymbol{\theta}) = \log h(\boldsymbol{x}) + T(\boldsymbol{x}) \cdot \boldsymbol{\theta} - A(\boldsymbol{\theta})$$

Hence we can directly relate negative log likelihood and Bregman divergence by

$$B_F(T(\boldsymbol{x})||a(\boldsymbol{\theta})) = -\log P(T(\boldsymbol{x})|\boldsymbol{\theta}) + \log h(\boldsymbol{x}) + F(T(\boldsymbol{x}))$$

□

Thus from a parameter estimation point of view, the minimization of Bregman divergence and the maximization of log likelihood are equivalent. Now consider an arbitrary time series $\{x_{1m}, x_{2m}, ..., x_{Tm}\}$ in the data set, whose marginal distribution (for each x_{mt}) belongs to some exponential family, a natural extension of the "fitness" loss is the Bregman divergence. Together with the above discussion, the first term in the proposed multiple time series smoothing formulation (3.5) could be generalized as

$$\begin{aligned} l(\Theta) &= \sum_{m=1}^{M}\sum_{t=1}^{T} B_F\left(T(x_{mt})||a(\theta_{mt})\right) \propto \sum_{m=1}^{M}\sum_{t=1}^{T} -\log P(T(x_{mt})|\theta_{mt}) \\ &\propto \sum_{m=1}^{M}\sum_{t=1}^{T} \{A(\theta_{mt}) - T(x_{mt})\theta_{mt}\} \end{aligned} \tag{3.18}$$

where we use the matrix $\Theta \in \mathbb{R}^{M\times T}$ to denote all natural parameters associated with the elements of the multiple times series. For commonly used exponential family distributions, their normalization factor $A(\theta)$, the corresponding $a(\theta)$, and the transformation to natural parameters can be found on the last table of https://en.wikipedia.org/wiki/Exponential_family. For example, with Gaussian assumption for continuous time series, $A(\theta) = \frac{\theta^2}{2}$ and $a(\theta) = \theta$, we have

$$B_F\left(T(x_{mt})||a(\theta_{mt})\right) \propto -x_{mt}\theta_{mt} + \frac{\theta_{mt}^2}{2} \propto (x_{mt} - \theta_{mt})^2$$

which recovers the fitness term of the formulation in last section. If the time series contain binary records $x_{mt} \in \{+1, -1\}$, we have $A(\theta) = log(1 + e^{\theta})$, $a(\theta) = \frac{1}{1+e^{-\theta}}$, and with some calculations we get

$$B_F\left(T(x_{mt})||a(\theta_{mt})\right) \propto \log\left(1 + e^{-x_{mt}\theta_{mt}}\right)$$

Since natural parameters uniquely characterize the exponential family distribution, in particular the moments through cumulant function, it appears reasonable to adopt a similar regularization as in (3.5) for natural parameters of each entry, to impose temporal smoothness on each time sequence, as well as their inter-correlations. As such, the overall learning objective of general multiple time series smoothing reads

$$\begin{aligned}\min_{\boldsymbol{\theta}_1,\cdots,\boldsymbol{\theta}_M} \mathcal{J}(\Theta) = &\sum_{m=1}^{M}\sum_{t=1}^{T}\{A(\theta_{mt}) - T(x_{mt})\theta_{mt}\} \\ &+ \lambda_1 \sum_{m=1}^{M}\sum_{t=2}^{T-2}\left(\nabla_t^2\theta_{mt}\right)^2 + \lambda_2 \sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M}\sum_{t=2}^{T-1}\left[\nabla_t^2(\theta_{it} - C_{ij}\theta_{jt})\right]^2\end{aligned} \tag{3.19}$$

which is still convex since the second order derivative of each component of the first term is $a'(\theta_{mt}) = \mathrm{Var}(T(x_{mt})) > 0$.

3.2.3 A Fast Random Block Coordinate Descent (RBCD) Algorithm

So far the problem of multiple time series smoothing has been reduced to solving a convex optimization problem (3.19) with smoothness penalty λ_1 and λ_2 as hyper-parameters. Generic methods, such as those based on first or second order gradient [116, 117], may be applied but may not be a good choice - the dimension of the decision variables Θ equals to the number of elements of all time series, hence the calculation or even the storage of full first/second order gradient is quite inefficient. Moreover, batch gradient methods suffers from the choice of step size and numerical instability when dealing with high-dimensional problems.

In this section, we propose a simple yet efficient algorithm that can be implemented in just a few lines of code. The key idea is the archetype of an universal solution methodology to algorithmic optimization: solving a complex or large scale problem by reducing it to a sequence of simpler optimization problems. More specifically for (3.19), it appears that fixing all the other decision variables except $\boldsymbol{\theta}_t$, (which are the decision variables corresponding to all observations of the multiple time series at time t), the sub-problem has low dimension and the solution can be updated easily with much less time and memory. We provide a convergence analysis of the proposed RBCD algorithm, and demonstrate its relation to stochastic gradient descent (SGD). In addition, RBCD is readily amendable for parallel computation, and empirically outperforms the state-of-the-art alternating direction method of multipliers (ADMM) that was recently proposed for total variation regularized problems [118, 119].

The RBCD start with an initial guess of the decision variables Θ^0. In each step, it consists of (1) picking up an index i_k from $\{1, \cdots, T\}$, (2) evaluating the gradient of a block of variables, i.e., $[\nabla\mathcal{J}(\Theta)]_{i_k}$ in the current implementation, followed by (3) updating the i_k^{th} column of Θ. Note that we have adopted the "subset indexing" convention: here and

throughout, $[\nabla\mathcal{J}(\Theta)]_i$ is used to denote the i^{th} column of $\nabla\mathcal{J}(\Theta)$. The indicator vector v_i has dimension $T \times 1$ and all its elements, except the i^{th} entry, equal to zero. The multiplication with v_i^T serves to match the dimension of block gradient to the dimension of all decision variables. Also it is worth pointing out that in each step i_k could be chosen randomly, as in the current implementation, for the purpose of parallel computing. Alternatively i_k can be selected in a deterministic fashion, e.g., using a cyclic schedule. The convergence analysis in later part of this section holds for both cases.

Algorithm 3: Random Block Coordinate Descent (RBCD) Algorithm

Input: Multiple time series $X = [\boldsymbol{x}_1, ..., \boldsymbol{x}_T] \in \mathbb{R}^{M\times T}$

Initialize $\Theta^0 = [\boldsymbol{\theta}_1^0, ..., \boldsymbol{\theta}_T^0] \in \mathbb{R}^{M\times T}$, and let $k \leftarrow 0$

while $k < iter_{max}$ **do**

  Sample $i_k \in \{1, \cdots, T\}$ from a uniform distribution

  $\Theta^{k+1} \leftarrow \Theta^k + \alpha_k[\nabla\mathcal{J}(\Theta)]_{i_k} v_{i_k}^T$

  if $||\Theta^{k+1} - \Theta^{k-T+2}|| < threshold$ **then**

    return

  $k \leftarrow k + 1$

Now we calculate the gradients that are required by the algorithm. To begin with, the three terms of the objective function (3.19) are denoted by $l(\Theta)$, $\Omega_1(\Theta)$ and $\Omega_1(\Theta)$, respectively, i.e., the objective function is rewritten as

$$\mathcal{J}(\Theta) = l(\Theta) + \lambda_1\Omega_1(\Theta) + \lambda_2\Omega_1(\Theta) \tag{3.20}$$

for clarity. When all elements of Θ except the i^{th} column $\boldsymbol{\theta}_i$ are fixed, we can easily compute

$$\frac{\partial l(\Theta)}{\partial \boldsymbol{\theta}_i} = -\left(a(\boldsymbol{\theta}_i) - T(\boldsymbol{x}_i)\right) \tag{3.21}$$

where the function operation should be interpreted component-wise, i.e.,

$$a(\boldsymbol{\theta}_i) \triangleq [a(\theta_{1i}), \cdots, a(\theta_{Mi})]^T$$

The gradient computation of the second term is also straightforward,

$$\frac{\partial\Omega_1(\Theta)}{\partial \boldsymbol{\theta}_i} = \phi(B)\boldsymbol{\theta}_i \tag{3.22}$$

where $\phi(B) = B^2 - 4B + 6 - 4B^{-1} + B^{-2}$ and B is the time delay operator. The gradient of the third term is more involved, with some algebra we get

$$\frac{\partial\Omega_2(\Theta)}{\partial \boldsymbol{\theta}_i} = \phi(B)\left[(M-3)I + 2C + \text{diag}\left(\sum_{j=1}^{M} C_{1j}^2, \cdots, \sum_{j=1}^{M} C_{Mj}^2\right)\right]\boldsymbol{\theta}_i \tag{3.23}$$

Finally the block gradient required in the algorithm can be obtained by combining the above three terms, i.e.,

$$[\nabla \mathcal{J}(\Theta)]_i = \frac{\partial l(\Theta)}{\partial \boldsymbol{\theta}_i} + \lambda_1 \frac{\partial \Omega_1(\Theta)}{\partial \boldsymbol{\theta}_i} + \lambda_2 \frac{\partial \Omega_2(\Theta)}{\partial \boldsymbol{\theta}_i} \tag{3.24}$$

Now we provide the convergence analysis of the algorithm.

Theorem 18. *The gradient function $\nabla \mathcal{J}(\Theta)$ is block-wise Lipschitz continuous. Let L_i be the Lipschitz constant of block i, then*

$$\begin{aligned} L_i &\geq (2 + 12\lambda_2 + 2\lambda_2(M-3)) + 2\|C\|_2 + \min\{\sum_{j=1}^{M} C_{1j}^2, \cdots, \sum_{j=1}^{M} C_{Mj}^2\} \triangleq \bar{L}_{min} \quad \forall i \\ L_i &\leq (2 + 12\lambda_2 + 2\lambda_2(M-3)) + 2\|C\|_{\mathcal{F}} + \max\{\sum_{j=1}^{M} C_{1j}^2, \cdots, \sum_{j=1}^{M} C_{Mj}^2\} \triangleq \bar{L}_{max} \quad \forall i \end{aligned} \tag{3.25}$$

The RBCD algorithm with constant step size $\alpha_k = \bar{L}$ generates a sequence $\{\Theta^k\}_{k\geq 0}$ that achieves

$$\mathbb{E}[\mathcal{J}(\Theta^k)] - \mathcal{J}^* \leq \left(1 - \frac{\bar{L}_{min}}{T\bar{L}_{max}}\right)^k (\mathcal{J}(\Theta^0) - \mathcal{J}^*) \tag{3.26}$$

Interestingly, the proposed RBCD method is closely related to the Stochastic Gradient Descent (SGD) method which has received much attention for large scale machine learning application. SGD tries to minimize a smooth function f by taking a negative step along an estimate g of the gradient $\nabla f(x)$. Under regular conventions, it is assumed that g is unbiased, i.e., $\mathbb{E}[g] = \nabla f(x)$, where the expectation is taken over the random variables that are used to obtain g at current value of x. The proposed RBCD method, somewhat surprisingly, can be viewed as a special case of the above SGD. In fact, if we take

$$g = T[\nabla \mathcal{J}(\Theta)]_{i_k} v_{i_k}^T,$$

then with the random sampling of the coordinate index, we have

$$\mathbb{E}[g] = \frac{1}{T}\sum_{i=1}^{T} T[\nabla \mathcal{J}(\Theta)]_i v_i^T = \nabla \mathcal{J}(\Theta) \tag{3.27}$$

3.3 A Contextual Bayesian Approach

In this section, we proceed to establish a probabilistic graphical model that models multiple correlated time series data. The key idea involves using matrix factorization based collaborative filtering to capture the relatedness among multiple time series, while taking the advantage of Hidden Markov model (HMM) for the modeling of temporal dependence. The construction of the model is not only closely related to dynamic system identification, but also has a graphical model representation shown in Figure 3.4. As before, we have denoted by $\boldsymbol{x}_t$ for the t^{th} column of the observation matrix X.

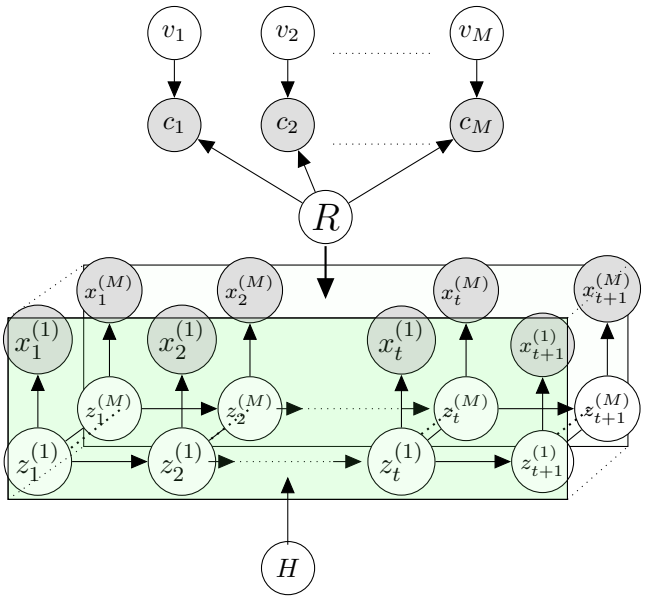


Figure 3.4: Graphical representation of Contextual Hidden Markov Model (CHMM)

3.3.1 Collaborative Filtering with HMM

To begin with, the observed multiple time series can be thought of as the measurement a dynamic system recorded by M sensors from time 1 to T, i.e., the m^{th} row of the data matrix $X \in \mathbb{R}^{M \times T}$ contains all readings from sensor m. Similar to the widely used low rank matrix decomposition for collaborative filtering [120], one can try to find two matrices $R \in \mathbb{R}^{M \times p}$ and $Z \in \mathbb{R}^{p \times M}$, such that

$$X_{M \times T} \simeq R_{M \times p} Z_{p \times T} \tag{3.28}$$

where R can be viewed as a "sensor latent" matrix, and Z as a "system latent" matrix. More specifically, each column of the above equation reads

$$\boldsymbol{x}_t \simeq R \boldsymbol{z}_t \tag{3.29}$$

from which one can treat $\boldsymbol{z}_t$ as the system hidden states, and R as the "observation matrix" in the terminology of linear system theory. Similarly, the correlations among multiple time series, encapsulated in the matrix C, can be decomposed with the help of a "relatedness latent" matrix $C \in \mathbb{R}^{p \times M}$, i.e.,

$$C_{M \times M} \simeq R_{M \times p} V_{p \times M} \tag{3.30}$$

Column-wise the above formula imposes

$$\boldsymbol{c}_j \simeq R_{M \times p} \boldsymbol{v}_j \tag{3.31}$$

So far all the decomposition described above only takes care of inter-series relatedness. To incorporate temporal dependence, we propose using HMMs to model X and Z, shown at

the bottom of Figure 3.4. The essence of HMM is to assume that there is a hidden Markovian process[2] $\boldsymbol{z}_t$ that drives the observation $\boldsymbol{x}_t$, in a manner that $\boldsymbol{x}_t$ depends instantaneously on $\boldsymbol{z}_t$. With the Markovian property, we only need to model $p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})$ and $p(\boldsymbol{x}_t|\boldsymbol{z}_t)$ for each steps. With further homogeneous assumption, the probabilistic models of all steps reduces to a single transitional distribution $p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})$ $\forall t$ and an emission distribution $p(\boldsymbol{x}_t|\boldsymbol{z}_t)$ $\forall t$. By the theory of d-separation in probabilistic graphical models, $p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \cdots, \boldsymbol{x}_1)$ cannot be further reduced using any conditional independence rule implied by HMM, hence the temporal dependence is "preserved". Moreover, the introduction of latent variables $\boldsymbol{z}_t$ empowers additional modeling flexibility for possibly non-stationary observations, which commonly shows up in real-world applications. From a dynamic system analysis viewpoint, the above HMM construction simply introduces an additional relation:

$$\boldsymbol{z}_{t+1} \simeq H\boldsymbol{z}_t \tag{3.32}$$

Overall, the aforementioned intuition suggests modeling the distributions of $p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})$ for temporal dependence, $p(\boldsymbol{x}_t|\boldsymbol{z}_t, R)$ for "observation transformation", and $\prod_{j=1}^{M} p(\boldsymbol{c}_j|\boldsymbol{v}_j, R)$ for inter-series relatedness. With that the complete likelihood of the proposed probabilistic model for multiple time series reads,

$$\begin{aligned} &l(X, Y, R, C, V) = \\ &p(\boldsymbol{z}_0) \underbrace{\prod_{t=1}^{T} p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})}_{\text{temporal dependence}} \underbrace{\prod_{t=1}^{T} p(\boldsymbol{x}_t|\boldsymbol{z}_t, R)}_{\text{observation}} \underbrace{\prod_{j=1}^{M} p(\boldsymbol{c}_j|\boldsymbol{v}_j, R)p(\boldsymbol{v}_j)}_{\text{context}} \end{aligned} \tag{3.33}$$

where we have added extra terms like $p(\boldsymbol{z}_1)$ and $p(\boldsymbol{v}_j)$ to incorporate the prior information of the corresponding random variables. The above likelihood formulation resembles an HMM model imposed with "emission constraints" from the contextual layer. Hence we call it Contextual HMM (CHMM). Again when Gaussian conventions from linear dynamics system analysis are adopted, one could substantiate each piece by an additive Gaussian model:

$$\begin{aligned} \boldsymbol{z}_t &= H\boldsymbol{z}_{t-1} + \alpha_t \\ \boldsymbol{x}_t &= R\boldsymbol{z}_t + \beta_t \end{aligned} \tag{3.34}$$

where $\alpha_t \sim \mathcal{N}(\boldsymbol{0}, \Lambda)$ and $\beta_t \sim \mathcal{N}(\boldsymbol{0}, \Xi)$ are i.i.d. multivariate Gaussian random variable with the same dimension of $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$, respectively. Similarly for the contextual layer one has

$$\boldsymbol{c}_j = R\boldsymbol{v}_j + \gamma_j \tag{3.35}$$

with i.i.d. random variables $\gamma_j \sim \mathcal{N}(\boldsymbol{0}, \Gamma)$. The prior distributions for $\boldsymbol{z}_0$ and $\boldsymbol{v}_j$ can be assumed to be Gaussian for conjugation, which yields $\boldsymbol{z}_0 \sim \mathcal{M}(\bar{\boldsymbol{z}}, \Upsilon_0)$ and $\boldsymbol{v}_j \sim \mathcal{M}(\boldsymbol{0}, \Phi_0)$.

[2] A process that satisfies the Markov Property, i.e., given current instance, the past and the future are independent.

3.3.2 EM Learning Algorithm

The CHMM proposed in the last section belongs to the more general probabilistic graphical model or Bayesian network framework. Due to the coupling of observations imposed by the contextual layer, the corresponding model learning (or parameter estimation) problem becomes much more challenging than classical HMM. Recall that in the current setting the available observations are measurements X and the correlation matrix C. The goal of model learning is to estimate model parameters, e.g., $\Theta = \{H, \Lambda, \Xi, \Gamma, \bar{\boldsymbol{z}}, \Upsilon_0, \Phi_0\}$, based on current observations. With the learned model at hand, one is ready to infer the distribution of latent variables, the expected observation value at certain temporal spatial locations.

While a wide variety of general graphical model learning methods, such as expectation-maximization (EM), variational methods, or sampling based approaches, exist in literature to cope with latent variables [121], In this section we establish a simple EM algorithm, based on the observation that conditioning on the "sensor latent" matrix R, the CHMM model decompses into a simple HMM and a matrix factorization. Before proceeding to detailed update formulas, it's useful to recall that each iteration of the usual EM algorithm consists of

- **Expectation step:** Under current estimate of the parameters Θ^k and observed data D, calculate the expected log likelihood function.

$$Q(\Theta|\Theta^k) = \mathbb{E}_{L|D,\Theta^k}[\log l(L, D; \Theta)] \tag{3.36}$$

  where L is the set of all latent variables. Usually the E-step can be reduced to computing the expectation of sufficient statistics.

- **Maximization step:** Maximize the above function to find

$$\Theta^{k+1} = \text{argmax}_\Theta Q(\Theta|\Theta^k) \tag{3.37}$$

  The key point is that in most cases the Q function is much easier to optimize, and in some cases can be solved explicitly as a function of sufficient statistics.

Regarding EM for CHMM, it's useful to realize that conditioning on the matrix R, the HMM layer and the contextual layer are independent. This inspires us to treat R as a model parameter, and then compute the expectations involved in the HMM layer and contextual layer separately in the E-step. Specifically, we reset model parameters as $\Theta = \{R, H, \Lambda, \Xi, \Gamma, \bar{\boldsymbol{z}}, \Upsilon_0, \Phi_0\}$, and latent variables $L = \{Z, V\}$. The EM algorithm for learning CHMM is summarized as follows.

In the E-step the model parameters Θ are assumed to be known (fixed with previous values). In the contextual layer, the expectation of latent variables can be easily obtained using the formula for the conditional distribution of multivariate Gaussian[3]. After some algebra we have

$$\boldsymbol{v}_j|\boldsymbol{c}_j \sim \mathcal{N}(B^{-1}R^T\boldsymbol{c}_j, \Gamma B^{-1}) \tag{3.38}$$

[3]For detailed property of Jointly Guassian, please refer to Chapter 4.3-4.3 of [121].

where $B = R^T R + \Phi^{-1}\Gamma$. Now we could easily compute

$$\begin{aligned}\mathbb{E}[\boldsymbol{v}_j|\boldsymbol{c}_j] &= B^{-1}R^T\boldsymbol{c}_j \\ \mathbb{E}[\boldsymbol{v}_j\boldsymbol{v}_j^T|\boldsymbol{c}_j] &= B^{-1}R^T\boldsymbol{c}_j\boldsymbol{c}_j^T RB^{-1} + \Gamma B^{-1}\end{aligned} \tag{3.39}$$

To update the expectations of the sufficient statistics of the latent variable Z, one just has to realize that given Θ, the bottom (HMM) layer of CHMM is reduced to a simple Gaussian linear system. Hence the E step of this block can be performed through standard Kalman Filer [122, 123, 124] for linear system smoothing:
-Forward propagation

$$\begin{aligned}\boldsymbol{\mu}_t &= H\boldsymbol{\mu}_{t-1} + K_t(\boldsymbol{x}_t - RH\boldsymbol{u}_{t-1}) \\ \Psi_t &= (I - K_t R)P_{t-1} \\ P_{t-1} &= H\Psi_{t-1}H^T + \Lambda \\ K_t &= P_{t-1}R^T(RP_{t-1}R^T + \Xi)^{-1}\end{aligned} \tag{3.40}$$

-Backward propagation

$$\begin{aligned}\hat{\boldsymbol{\mu}}_t &= \boldsymbol{\mu}_t + Q_t(\hat{\boldsymbol{\mu}}_{t-1} - H\boldsymbol{\mu}_t) \\ \hat{\Psi}_t &= \Psi_t + Q_t(\hat{\Psi}_{t+1} - P_t)Q_t^T \\ Q_t &= \Psi_t H^T(P_t)^{-1}\end{aligned} \tag{3.41}$$

Then we have the expectations for those sufficient statistics reads:

$$\begin{aligned}\mathbb{E}[\boldsymbol{z}_t] &= \hat{\mu}_t \\ \mathbb{E}[\boldsymbol{z}_t\boldsymbol{z}_{t-1}^T] &= \hat{\Psi}_t Q_{t-1} + \hat{\boldsymbol{\mu}}_t\hat{\boldsymbol{\mu}}_{t-1}^T \\ \mathbb{E}\boldsymbol{z}_t\boldsymbol{z}_t^T] &= \hat{\Psi}_t + \hat{\boldsymbol{\mu}}_t^T\end{aligned} \tag{3.42}$$

The goal of the M-step is to find the optimizer

$$\Theta^{k+1} = \text{argmax}_\Theta Q(\Theta|\Theta^k) \tag{3.43}$$

Fortunately, under the assumption of CHMM the optimization is concave and smooth. Thus it can be solved analytically by first order conditions. Setting derivatives to zero for each parameters, we obtain parameter updating formula as follows.
-initial/prior parameters

$$\begin{aligned}\bar{\boldsymbol{z}} &= \mathbb{E}[\boldsymbol{z}_1] \\ \Upsilon_0 &= \mathbb{E}[\boldsymbol{z}_1\boldsymbol{z}_1^T] - \mathbb{E}[\boldsymbol{z}_1]\mathbb{E}[\boldsymbol{z}_1^T] \\ \Phi_0 &= \frac{1}{M}\sum_{j=1}^{M}\mathbb{E}[\boldsymbol{v}_j\boldsymbol{v}_j^T]\end{aligned} \tag{3.44}$$

-transition paramters

$$
\begin{aligned}
H &= \left(\sum_{t=2}^{T} \mathbb{E}[\boldsymbol{z}_t \boldsymbol{z}_{t-1}^T]\right)\left(\sum_{t=2}^{T} \mathbb{E}[\boldsymbol{z}_{t-1} \boldsymbol{z}_{t-1}^T]\right)^{-1} \\
\Lambda &= \frac{1}{T-1}\sum_{t=2}^{T}(\mathbb{E}[\boldsymbol{z}_t \boldsymbol{z}_t^T] - H\mathbb{E}[\boldsymbol{z}_{t-1}\boldsymbol{z}_t^T] - \mathbb{E}[\boldsymbol{z}_t \boldsymbol{z}_{t-1}^T]H^T + H\mathbb{E}[\boldsymbol{z}_{t-1}\boldsymbol{z}_{t-1}^T]H^T)
\end{aligned}
\tag{3.45}
$$

-observation paramters

$$
\Xi = \frac{1}{T}\sum_{t=1}^{T}\left(R\mathbb{E}[\boldsymbol{z}_t \boldsymbol{z}_t^T]R^T + \boldsymbol{x}_t \boldsymbol{x}_t^T - 2I \circ R\mathbb{E}[\boldsymbol{z}_t]\boldsymbol{x}_t\right) \tag{3.46}
$$

-contextual paramters

$$
\begin{aligned}
R_{i,:} &= D_1 D_2^{-1} \\
\Gamma &= \frac{1}{M}\sum_{j=1}^{M}\left(R\mathbb{E}[\boldsymbol{v}_j \boldsymbol{v}_j^T]R^T \boldsymbol{c}_j \boldsymbol{c}_j^T - 2I \circ R\mathbb{E}[\boldsymbol{v}_j]\boldsymbol{c}_j\right)
\end{aligned}
\tag{3.47}
$$

where each row of the matrix R is updated by reweighting contributions of the contextual layer and HMM layer.

$$
\begin{aligned}
D_1 &= \rho\Gamma^{-1}\sum_{j=1}^{n} C_{ij}\mathbb{E}[\boldsymbol{v}_j^T] + (1-\rho)\Xi^{-1}\sum_{t-1}^{T} x_{it}\mathbb{E}[\boldsymbol{z}_t^T] \\
D_2 &= \rho\Gamma^{-1}\sum_{j=1}^{n} \mathbb{E}[\boldsymbol{v}_j \boldsymbol{v}_j^T] + (1-\rho)\Xi^{-1}\sum_{t-1}^{T} \mathbb{E}[\boldsymbol{z}_t \boldsymbol{z}_t^T]
\end{aligned}
\tag{3.48}
$$

Following standard complexity analysis of Kalman filter [125], each iteration of the EM procedure has time complexity $\mathcal{O}(M^3T)$ and space complexity $\mathcal{O}(M^2T)$. As a special case of the more general EM framework, the convergence analysis of the above procedure follows classical works [126, 127, 128]. The readers are also referred to recent works like [129, 130, 131] for a discussion on issues of convergence rate and online learning possibilities.

3.4 Experiment

This section is devoted to the verification of the proposed algorithms proposed, as well as discussing possible procedures for the choice of model hyperparameters. Overall, we will demonstrate, through a real-world applications, that the proposed multiple time series analysis tools enables the discovery of network level outliers/novelties that may otherwise be ignored by traditional single time series analysis methods.

3.4.1 Data Collection from a PMU network

The data-set used in this section was collected from a power distribution system equipped with smart meters called phasor measurement units (PMUs) [132]. Each channel of a particular PMU generates a time series by measuring one type of system state at a certain node. Figure 3.5 illustrates one of our case studies, in which five PMUs are installed at different locations in a distribution subsystem (top left), providing measurements of voltage/current magnitude and phase angle at a high sampling rate (bottom left). Due to the innate smoothness of state transition, the time series exhibits a strong correlation among adjacent measurements along the temporal dimension, as can be observed in the top right subfigure. Additionally, since all PMUs are connected with one another through the underlying power distribution network, the measurements also demonstrate non-negligible inter-series correlations, in particular for times series generated from the same branch of the network.

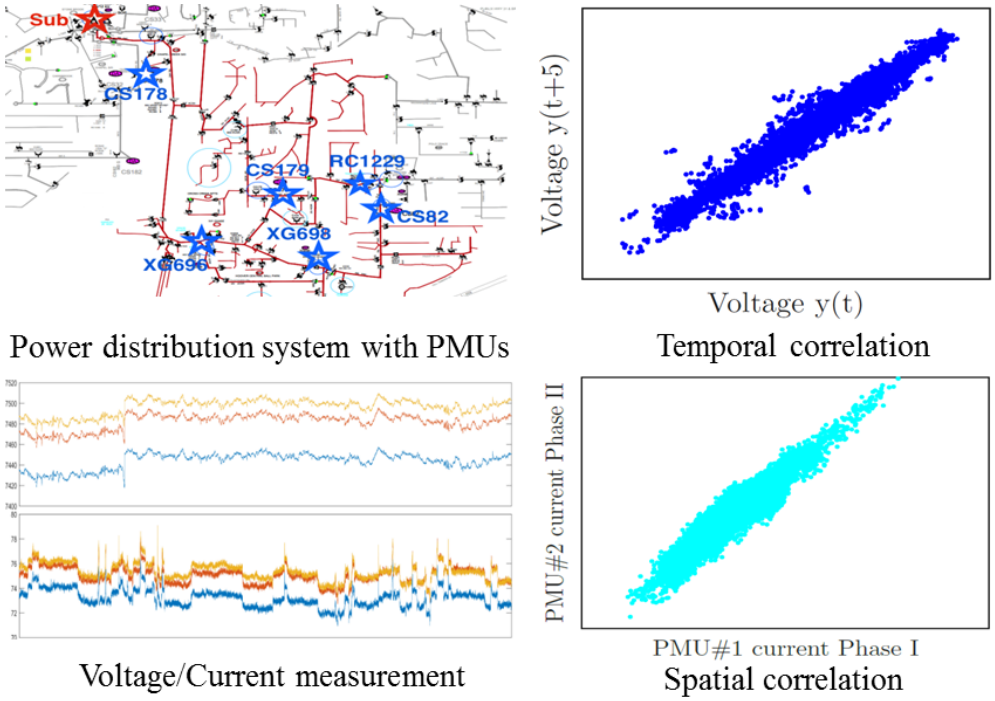


Figure 3.5: power distribution system equipped (PMUs) (top left); Voltage, current measurement of one PMU (bottom left); Temporal correlation with 5 steps delay (top right); Spatial correlation between current channels of PMU1 and PMU2 (bottom right).

All measurements from the PMU netowrk are GPS time stamped to provide time-synchronized observability. The smart meters used in this project provide three-phase voltage and current magnitude and phase angle with a 0.05% Total Vector Error and 20 seconds time resolution. Measurement data is collected during the period June 02 to July 11, 2015. Each sample is a 60 dimensional vector containing 12 channels per μPMU measuring three phase

voltage/current magnitude/angle. Thus for the mutiple time series model, the observed measurement X is $60 \times T$, and the empirical correlation matrix C has dimension 60×60.

3.4.2 Choice of Hyper-Parameters

The proposed non-parametric method has two hyperparameters λ_1 and λ_2, which are weights for temporal and inter-series smoothness, respectively. The proposed CHMM has one hyperparamter, p, which is the dimension of the hidden state Z. These hyperparameters determines the complexity of the learned model, and are critical for the performance of the two method. In the sequel we discuss the choice of hyperparameters within a cross validation (CV) framework.

First of all, a clean chunk of the multiple time series data[4] is randomly divided into training and testing sets. Let $B \in \mathbb{R}^{M \times T}$ be the indicator matrix having the same dimension as the data matrix X, i.e., $B_{ij} = 1$ if X_{ij} belongs to the training set, and $B_{ij} = 0$ if X_{ij} is assigned to the testing set. Each entry of B follows a Bernoulli distribution Ber(0.7), i.e., we use approximately 70% of the data for training and leave 30% for testing. Fortunately, both of the proposed methods are readily amendable to handle missing values (the data points held out for testing). For the non-parametric method, one can simply ignore the loss terms of the testing data points in the first part of (3.5), or more compactly, use $X \circ B$ to replace the first part of (3.7). Similarly, when the testing data points are held out, the E-step of the CHMM becomes a intermittent Kalman Filter [133], and the only modification needed for the M-step is to replace formula (3.48) with

$$\begin{aligned} D_1 &= \rho\Gamma^{-1} \sum_{j=1}^{n} C_{ij}\mathbb{E}[\boldsymbol{v}_j^T] + (1-\rho)\Xi^{-1} \sum_{t-1}^{T} B_{ij}x_{it}\mathbb{E}[\boldsymbol{z}_t^T] \\ D_2 &= \rho\Gamma^{-1} \sum_{j=1}^{n} \mathbb{E}[\boldsymbol{v}_j\boldsymbol{v}_j^T] + (1-\rho)\Xi^{-1} \sum_{t-1}^{T} B \circ \mathbb{E}[\boldsymbol{z}_t\boldsymbol{z}_t^T] \end{aligned} \tag{3.49}$$

To evaluate the CV performance, we use the root mean square error (RMSE) on the testing data set, i.e.,

$$\text{RMSE} = \sqrt{\frac{\left\|(1-B)\circ(X-\hat{X})\right\|_{\mathcal{F}}^2}{\sum_i \sum_t (1-B_{ij})}} \tag{3.50}$$

Figure 3.6 shows the impact of the two hyperparameters, λ_1 and λ_2, on the testing RMSE of the nonparameteric method. The 2D surface reaches a minimum when $\lambda_1 = 39$ and $\lambda_2 = 10$, demonstrating a trade-off between training fitness and smoothness (complexity). Based on that, we set the two weights accordingly for the nonparametric method.

[4]The data used for CV contains very few outliers and is different from the chunk of data used in the next section for validation (evaluation of the two methods).

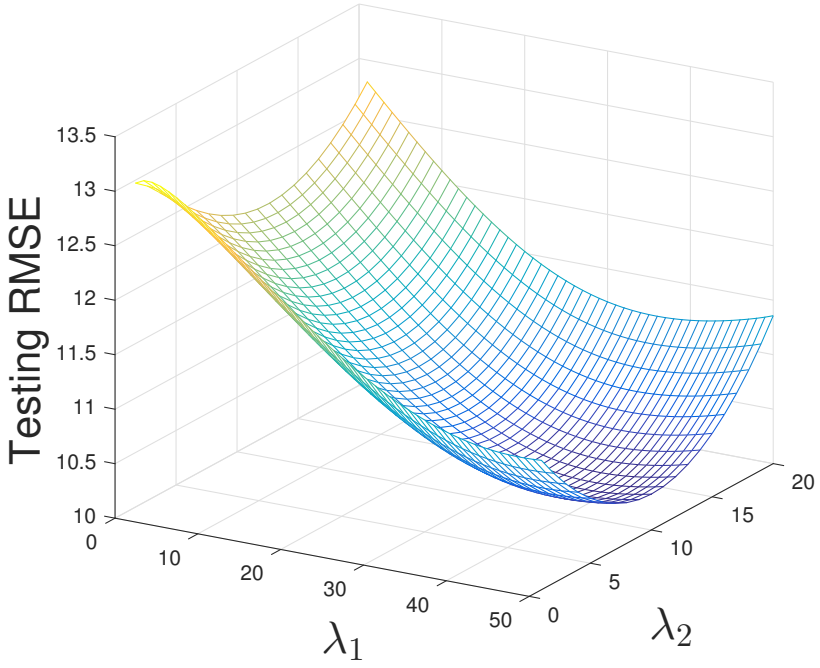


Figure 3.6: The testing RMSE of the non-parametric method as a function of hyperparameters

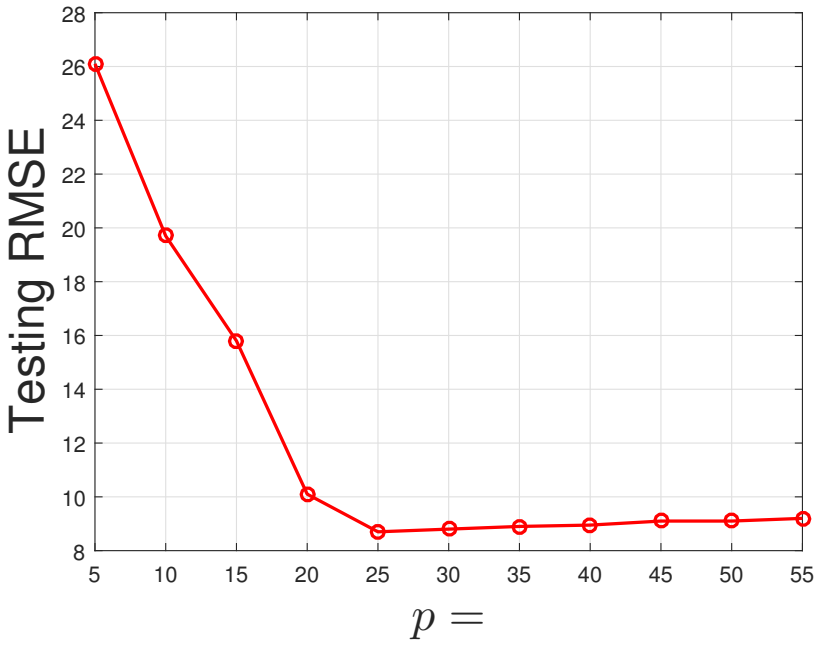


Figure 3.7: The testing RMSE of the CHMM method as a function of hyperparameters

Figure 3.7 shows the the testing RMSE of the CHMM method as a function of the

hidden state dimension p. It is seen that the RMSE decreases drastically in the first few steps, reaches a minimum at around $p = 25$, and increases slightly as p gets larger. This is understandable as p also characterizes the complexity (low rank approximation) of the CHMM model. Hence in the evaluation phase, we set $p = 25$.

3.4.3 Outlier and Novelty Detection Results

Next we test the proposed methods as a tool for outlier or novelty detection. A 120 minute measurement sequence is taken out, which exhibits abnormalities due to sensor or communication failure, and novel events like voltage disturbance due to load changes. Since outliers/novelty are defined as data points that deviate from the expected values under normal operation. we compute an index of novelty by comparing the inferred values $\widehat{X}_{it}$ with the observed values of X_{it} with the absolute distance.

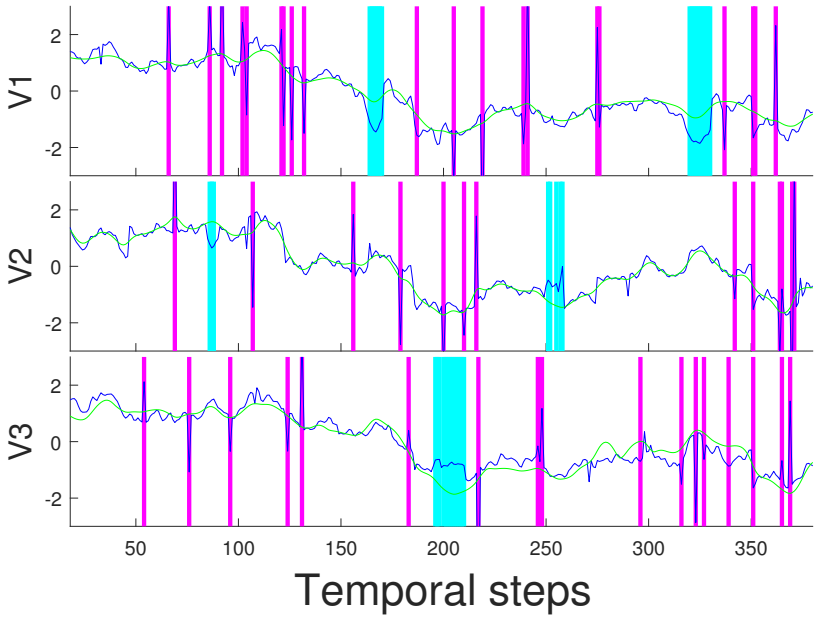


Figure 3.8: Outlier/Novelty Detection with the proposed multiple non-parametric method.

Figure 3.8 shows the detection results of the proposed non-parametric method. Note that although data from all 60 pmus/channels are used, only three correlated voltage streams are shown here for clearer presentation. The blue curve in each subplot is the raw data with outliers, and the green curve is the estimated values with the non-parametric method. It is seen that the estimated values are smoothed version of the original data and the measurement noise has been canceled out. For each time series, outliers/novelties are marked with vertical lines when the absolute difference between raw value and estimated value is larger than 0.73, which is 2σ calculated from all estimation biases. It appears that our method successfully

captured almost all outliers caused by sensor/communication problems or load changes. Those outliers are marked in magenta in each of the panel.

More interestingly, due to the incorporation of inter-series dependence, the estimated values for each time series do not always follow its own trend, but are also influenced by other correlated time series. This feature enables the detection of "network level" outliers and novelties, i.e., those data points that significantly violate the correlation structure of the system under measurement. This type of outliers are marked in cyan in each panel of Figure 3.8. Intuitively, they correspond to power grid events such as three phase imbalance, real and reactive power switching, etc.

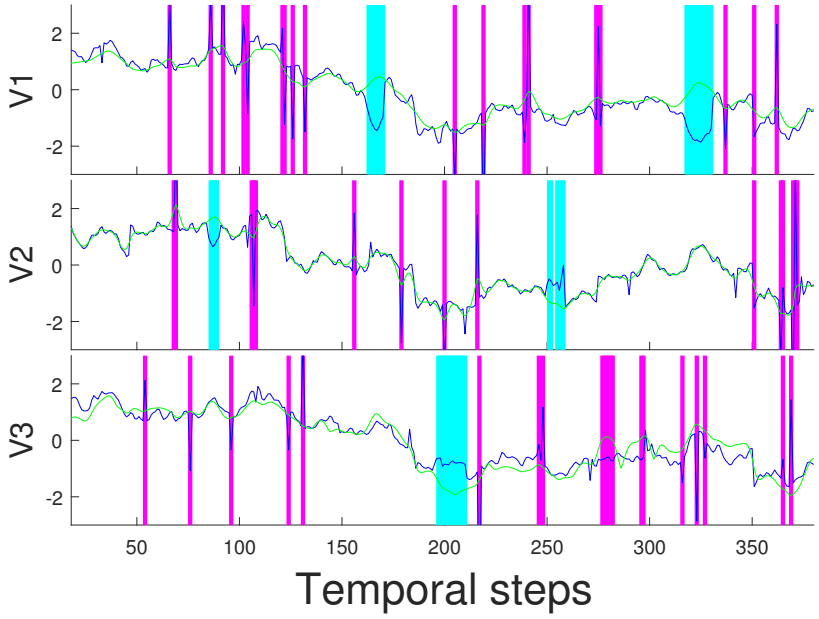


Figure 3.9: Outlier/Novelty Detection with the proposed CHMM method.

Figure 3.9 shows the detection results with the proposed CHMM method. At a first glance, the estimated values (green curves) are quite similar to those based on the non-parametric method. In general, CHMM also successfully detects both single stream and network level outliers. The only difference, compared to the non-parametric method, is that CHMM seems to emphasize inter-series relatedness more, while the temporal trend of each series is weighted less. This is understandable as the non-parametric method directly enforces smoothness while CHMM only tries to model this dependence through an HMM.

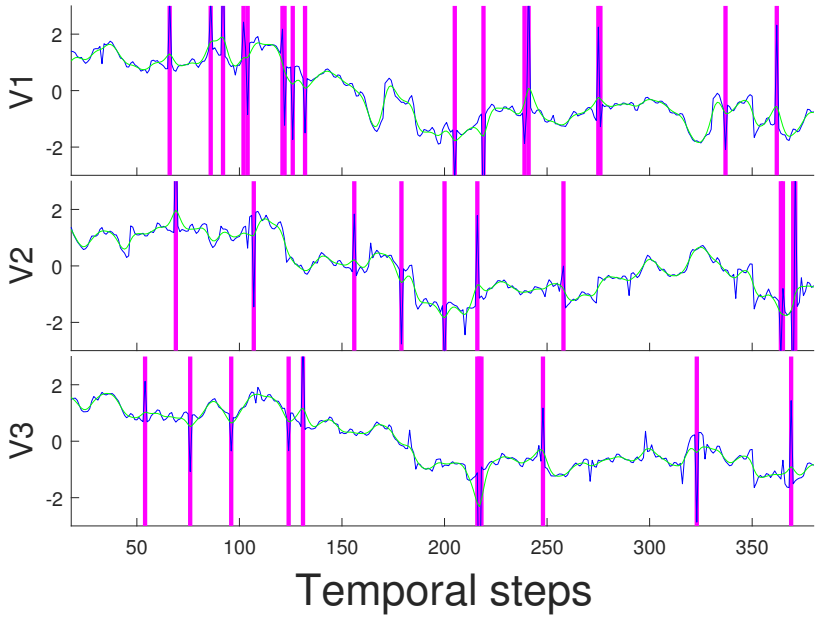


Figure 3.10: Outlier/Novelty Detection using single non-parametric modeling method.

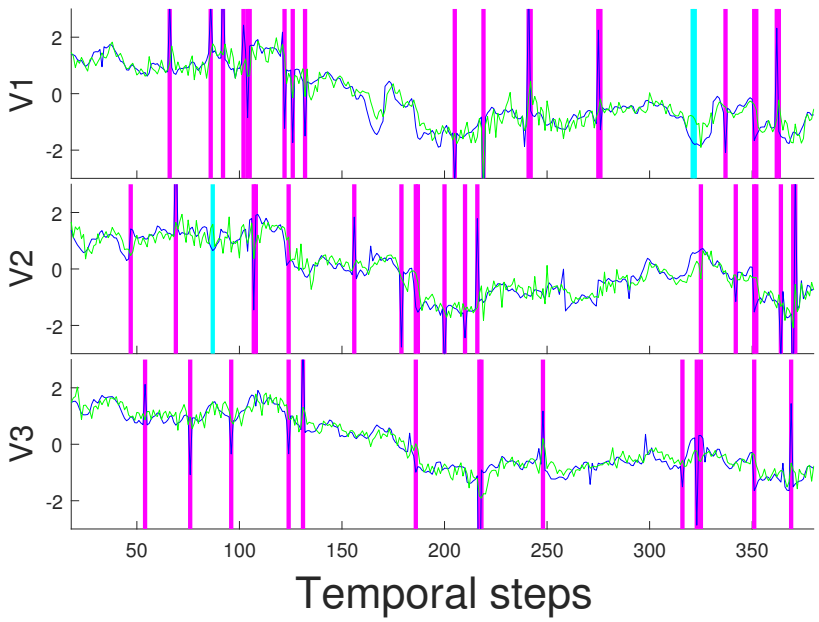


Figure 3.11: Outlier/Novelty Detection using multivariate ARIMA method.

To further justify the proposed two methods and the benefits of incorporating inter-series dependence, we compare them with two alternatives: One is the smoothing spline method [134] for each time series, and the other is the multivariate auto-regressive integrated moving average (mARIMA) model [135]. Note that the model selection (smoothing parameter

selection) of the spline method follows a similar cross validation introduced in the last section. The model selection of mARIMA, on the other hand, follows the procedure discussed in Chapter 4.3 of [135]. The outlier detection results for the spline method and mARIMA model are shown in Figure 3.10 and Figure 3.11, respectively. Apparently, the single task spline method fails to detect outliers that violate the correlation structure, although in general it provide well-fitted trend for each sequence. The detection results of mARIMA are interesting: due to the non-stationary nature (even after taking difference) of the measurement data, ARIMA model does not provide a good estimation in general. It is observed that some of the single stream outliers were missed, although the method is able to detect several network level outliers.

3.4.4 Empirical Evaluation of Computational Cost

Here we empirically test the computational cost of the RBCD and EM learning algorithms established for the non-parametric method and CHMM, respectively. The convergence of the RBCD algorithm is shown in Figure 3.12, where the y-axis is the value of the objective function (3.5), and x-axis is the number of iterations. The RBCD algorithm converges in $18T$ iterations, and in each iteration an explicit update is performed for a randomly selection column of the non-parametric model. The result justifies the theoretical convergence analysis. More interestingly, we see that although RBCD can be viewed as a special form of SGD, it is different from traditional SGD in that in each iteration a decrease of the objective function is guaranteed. The convergence result of the EM algorithm for CHMM is presented in Figure 3.13. Note that the objective is the negative likelihood, and we see that EM converges in about 67 iterations.

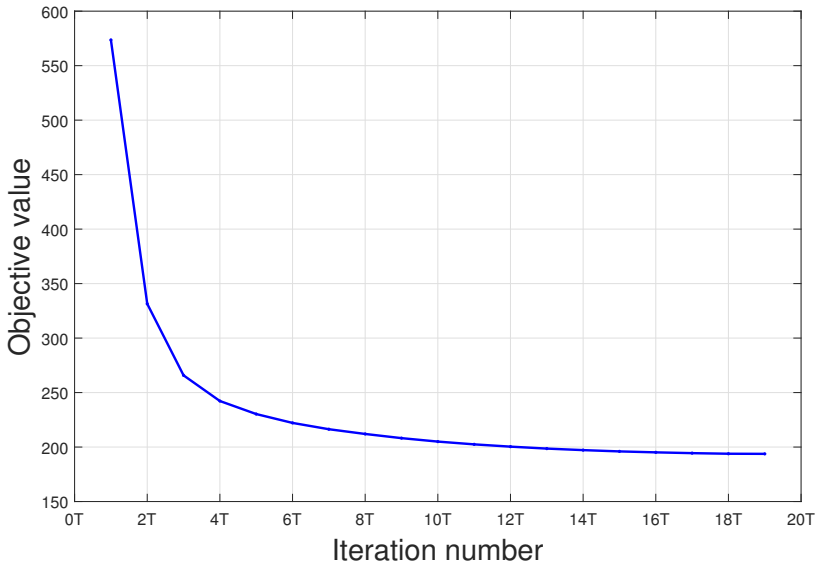


Figure 3.12: Convergence of the RBCD algorithm for the non-parametric method.

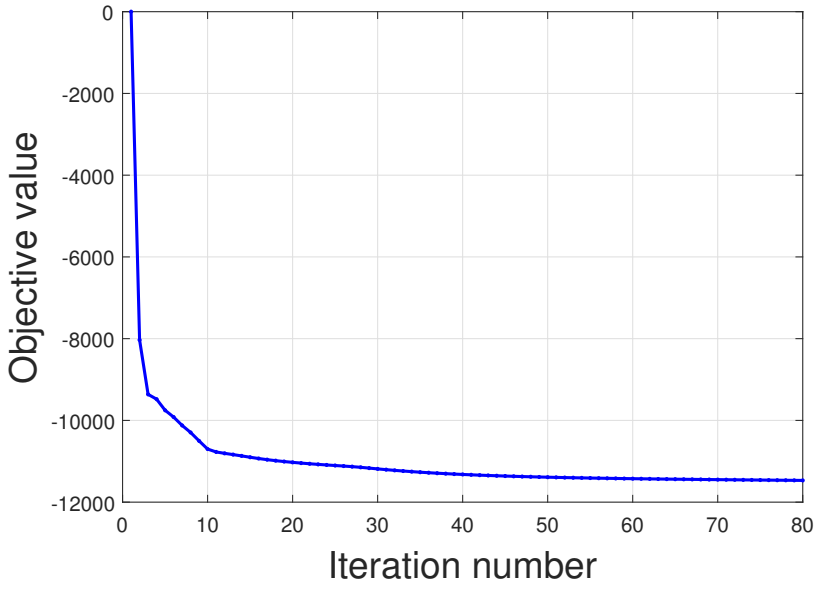


Figure 3.13: Convergence of the EM algorithm for CHMM.

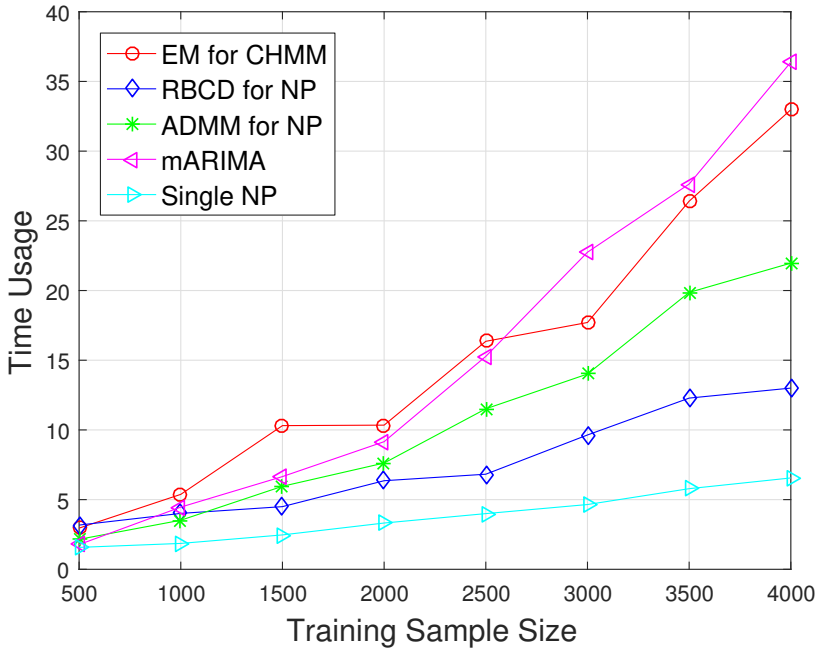


Figure 3.14: Comparison of Time Usage.

We also compare the computational cost of RBCD for the non-parametric method and EM for the CHMM, with mARIMA and signal-series non-parametric. To justify the benefit of RBCD, we also include the popular Alternating Direction Method of Multipliers (ADMM)

algorithm [118], for the parallel optimization of (3.5). All numerical experiments are performed on a workstation having dual Xeon5687 CPUs and 72GB memory. The results shown in the sequel are average values of 20 repetitions. Figure 3.14 illustrates the required computational time as a function of increasing size of training sequences. Among all methods that incorporate inter-series relatedness, RBCD-NP is the most efficient: For large training size it significantly reduces the running time by at least 42.1% compared to the runner-up ADMM-NP. Although single-NP takes the least time usage, its detection performance is poor and it misses all network level outliers, as is seen in last section. It appears that the computational costs of EM-CHMM and mARIMA scale slightly super-linearly and are both much more expensive than that of the non-parametric method. One advantage of EM-CHMM, however, is that its model selection is easier: CHMM only requires the specification of the hidden state dimension p, which can be chosen with a simple discrete line search in the CV framework.

3.4.5 Missing Value Recovery

Besides outlier and novelty detection, the proposed two methods in this chapter can also be used to recover missing values in the multiple time series data, as is detailed in Section 3.4.2. In this subsection, we evaluate the performance of missing value imputation on a 1 day (1440 mins) data set collected from the PMU network. 30% of the data points are held out randomly as missing values, and the RMSE metric defined in (3.50) is used for comparison purposes.

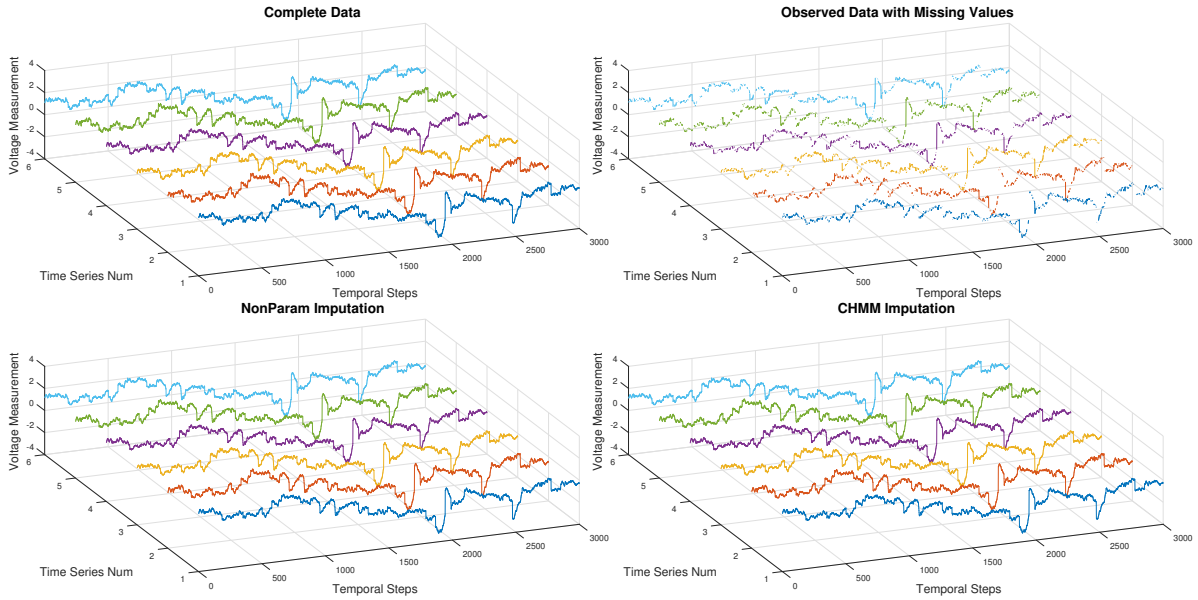


Figure 3.15: Missing Value Recovery with the Proposed Methods.

Figure 3.15 shows the imputation results obtained with the proposed two methods. The

top left panel shows the original data and the top right panel is the data with missing values. Note that in each of the subplot only 6 correlated time series are shown for illustration clarity, but the experiment is conducted for all 60 series. The recovery results of the non-parametric method and CHMM are presented in the bottom left and bottom right panels, respectively. It is observed that both methods are able to leverage temporal and inter-series dependence to achieve reasonable recovery.

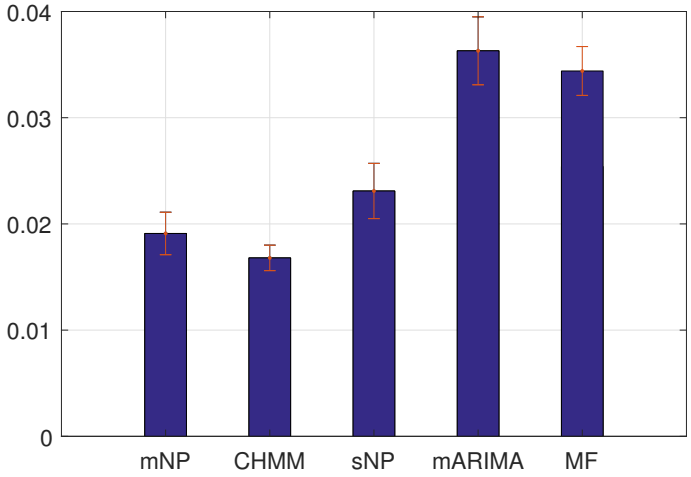


Figure 3.16: Comparison of RMSE of different missing value imputation methods

Finally, the imputation performance of the proposed methods is compared to three alternatives, including single stream non-parametric spline method (sNP) [134], multivariate ARIAM (mARIMA) [135], and collaborative filtering based on matrix factorization (MF) [120]. The testing RMSEs of all methods are shown in Figure 3.16. The mean RMSE and its confidence intervals $\pm 2\sigma$ are calculated from 20 repetitions of the randomized experiment. We observe that the CHMM outperforms all the other methods, and the runner-up, the proposed multiple time series non-parametric method (mNP), is also quite competitive. Overall, both CHMM and mNP outperforms traditional methods by at least 21% in terms of RMSE.

Chapter 4

Learning Operational Domain for Agile Control

This chapter addresses the problem of "machine learning for system operation". In particular, we focus on tasks including the learning of convex function for optimal control, consensus region for operational domain description, and semi-supervised classifier for event identification. For each of the three problems under consideration, we first discuss its learning formulation and provide generalization analysis. Then we establish a novel optimization scheme, namely parametric dual maximization (PDM), to solve the non-convex learning problem. PDM reveals an interesting "piece-wise convex" structure of a class of machine learning problems, including not only the above three problems but also various versions of semi-supervised learning, learning with hidden structures, robust learning, etc. By leveraging that structure and deriving a local explicit form of the solution, PDM essentially transforms these problems into *convex maximization*, and uses the idea of level set to approach global optimality. Finally, we conduct numerical experiments to demonstrate: (1) The performance of the proposed PDM procedure, compared to the state-of-the-art methods like stochastic gradient descent (SGD), concave-convex procedure (CCCP), block alternating optimization (BAO), branch and bound (B & B), etc. (2) The effectiveness of the three learning scheme and their usage in real-world system operation applications.

4.1 Learning Convex Functions for Optimal Control

4.1.1 Motivation and Formulation

Optimal control (OP) and its variations have achieved a great success and are becoming a common practice for applications ranging from classical problems such as trajectory tracking, vehicle control, to recent ones like manufacturing process control, economic mechanism design, energy system scheduling, etc [136] [137]. OP finds control policies for a system such that a desired criterion or an objective is optimized, given that system dynamics and opera-

tion requirements are satisfied throughout the control horizon. A typical optimal control in discrete time can be formulated as follows:

$$\min J = \Phi(x_0, T_0, x_f, T_f) + \sum_{t=T_0}^{T_f} \phi(x_t, u_t, t) \tag{4.1}$$

$$\text{s.t.} \quad x_{t+1} = f(x_t, u_t, t) \tag{4.2}$$

$$\varphi(x_0, T_0, x_f, T_f) = 0 \tag{4.3}$$

$$\rho(x_t, u_t, t) \leq 0 \tag{4.4}$$

where the functions $\Phi(\cdot)$ and $\phi(\cdot)$ in the objective specify the end point and process cost, receptively. The first equality constraint incorporates the system dynamics and the second constraint imposes a fixed initial or final state. The last constraint requires that the system operates within a feasible set, which is denoted as $\mathcal{A}_t$ and is called the acceptable set or the operation region of the system. Essentially, with system models established beforehand OP reduces to solving the above optimization problem.

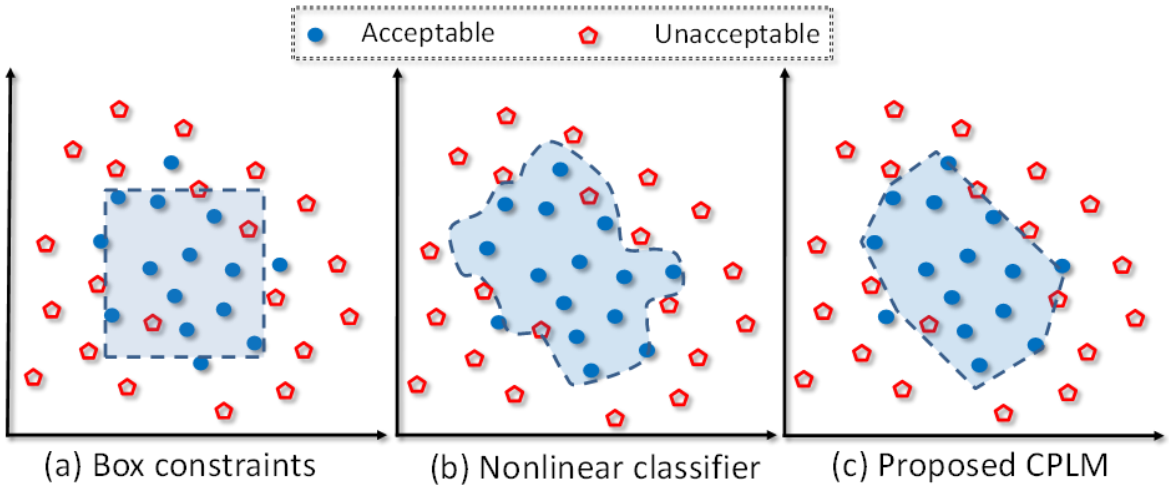


Figure 4.1: Different ways of describing operation requirement

In the traditional control scheme, the above formulation is substantiated through user specified objectives and physical laws that describe the relations among system variables. However, as recently OP is being applied to more complicated, stochastic, and human involved cyber physical systems, some of the relations might be hard to establish only with those "direct" methods. By exploiting the development of information technology and artificial intelligence, this difficulty can be alleviated by the so called data-driven approach in which measurements of the system are collected and machine learning tools are used to infer one or more system characteristics. Perhaps the most widely used data-driven method for

control is the estimation of system dynamics, known as system identification (SI) techniques [138] [139]. Nonetheless, there are very few research addressing the problem of learning operation constraints. In current literature only the ranges for each variable (box constraints) are considered [140] [141]. The over-simplified model is understandable from a technical standpoint: since the OP already requires solving a challenging large scale optimization problem, the incorporation of any complex operation region will induce non-linearly coupled constraints within the state variables, making the corresponding optimization very hard, if not intractable to solve [142]. The above concern also rules out many existing learning tools that construct non-linear classifiers, such as neural network, kernel SVM, or logistic regression, for the purpose of learning operation regions in the OP framework.

The solution proposed here follows the "learning for application" ideology, in which a learning machine is justified not only by its classification performance, but also its compatibility with the downstream applications. We suggest building Convex Piecewise Linear Machine (CPLM) for the modeling of system operation region. The advantage is obvious for the OP part: as CPLM is basically a set of linear inequalities, it can be directly plugged into any optimization without increasing the inherent complexity of the problem. A comparison of box constraints, non-linear classifiers and the CPLM is illustrated in Figure 4.1.

From a statistical learning viewpoint, the proposed CPLM tries to find an optimal configuration of multiple hyperplanes for classification. It is worth noting that in literature several attempts were made to learn piecewise linear classifiers: With additional assumption that some samples in the negative class have explicit subclass labels, [143] proposed an alternating method to learn polyhedrons. In [144] and [145] the author proposed another logistic formulation and a corresponding perception algorithm. Recently, the authors of [146] proposed a large margin formulation and a SGD algorithm to learn convex polytope machine. In the following, we formulate the learning problem by directly constructing the classifier from multiple hyperplanes and then proceed with regularized empirical risk minimization.

Let $\boldsymbol{x} \in \mathbb{R}^p$ be the p dimensional state variables of the system under consideration. Our goal is to build convex piecewise linear constraints to describe the operation region $\mathcal{A}$. Let $\mathcal{S} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{l}$ be the available training data set of size l, and $y \in \{+1, -1\}$ the corresponding label indicating $x \in \mathcal{A}$ or not. As is shown in Figure 4.1 (c), a convex set $\widehat{\mathcal{A}}$, defined by the intersection of M linear inequalities, is used to approximate the true $\mathcal{A}$, i.e.

$$\widehat{\mathcal{A}} = \left\{\boldsymbol{w}_1^T \boldsymbol{x} > 0\right\} \cap \cdots \cap \left\{\boldsymbol{w}_M^T \boldsymbol{x} > 0\right\} \tag{4.5}$$

$$= \left\{ \min_{1,\cdots,M} \{\boldsymbol{w}_1^T \boldsymbol{x}, \cdots, \boldsymbol{w}_M^T \boldsymbol{x}\} > 0 \right\} \tag{4.6}$$

where $\boldsymbol{w}_j$ is the parameter (normal vector pointing to the interior of the acceptable region) of the j^{th} hyperplane. To include interception one can simply add a dimension of constant 1 to $\boldsymbol{x}$. The equality (4.6) follows directly from a set logic argument. Thus CPLM is just the sign of the function

$$g(\boldsymbol{x}) = \min_{1,\cdots,M} \{\boldsymbol{w}_1^T \boldsymbol{x}, \cdots, \boldsymbol{w}_M^T \boldsymbol{x}\} \tag{4.7}$$

The M hyperplanes can be considered as M "experts". With the above decision rule a sample $\boldsymbol{x}'$ is classified as positive ($\hat{y} = +1$) if and only if all of the experts agree ($\boldsymbol{w}_j^T \boldsymbol{x} > 0 \ \forall j$), while a negative assignment is made as long as one expert "negate" ($\boldsymbol{w}_j^T \boldsymbol{x} > 0 \ \exists j$). Hence CPLM can be viewed as a "veto" combination of linear decision rules that emphasize the the sensitivity to the negative class and the specificity to the positive class.

As CPLM is a composed version of M linear hyperplanes, a natural concern is its generalization property, i.e. how well in theory the classifier will perform on unseen data, which is also of great importance for model selection. In the Probably Approximately Correct (PAC) learning framework, this problem reduces to analyzing the trade-off between training fitness and model complexity. Here we adopt the well known VC-dimension, denoted by d, as the measure of complexity. Intuitively the ways of shattering $\boldsymbol{x}$ with g is directly related with the dimensionality p and the number of hyperplanes M used in constructing the classifier, hence the VC-dimension of CPLM should be a function of both. While for two dimensional case ($p = 2$) the VC-dimension is known to be $2M + 1$ [147], for higher dimensional cases direct calculation becomes very difficult. As a detour, by using a geometric argument we get the following lower and upper bound for all p

Lemma 19. $pM \leq d \leq 2(p+1)M \log_2 [(p+1)M]$

For clarity all proofs are moved to appendix. The lower bound implies that the class of convex piecewise linear functions g has considerable description abilities (complexity) in high dimensions with large M, and the upper bound shows that the dependence on M is no larger than $\mathcal{O}(M \log M)$. With this, we have

Theorem 20. *Denote* $d' \triangleq 2(p+1)M \log_2 [(p+1)M]$, $R(g)$ *the generalization risk (0-1 loss) and* $\widehat{R}(g)$ *the empirical risk. Assume large enough sample size* $l > d'$, *we have that with probability at least* $1 - \delta$

$$R(g) \leq \widehat{R}(g) + \sqrt{\frac{2 \log(el/d')}{l/d'}} + \sqrt{\frac{\log(1/\delta)}{2l}}$$

The first term $\widehat{R}(g)$ in the above upper bound is the training cost (fitness), and the second term is a function of model complexity. With large enough l, the last term goes to 0 and the second term is a increasing function of d'. The bound is an instance of Occam's razor principle that when similar training costs are induced by a group of classifiers, simpler ones are preferred since they are more likely to generalize better. In the view of lemma 4.5.1 and theorem 4.2.1, a large number of hyperplanes is depreciated especially when CPLM is used for high dimensional data set. Moreover, the monotonicity with M suggests a simple incremental cross validation method for the choice of the number of hyperplanes.

4.1.2 Cost Sensitive Large Margin Learning Formulation

The next step is to learn CPLM from data. A generic learning task can be formulated as minimizing

$$L = r(f) + \sum_{i=1}^{l} l_i(y_i, f(\boldsymbol{x}_i)) \tag{4.8}$$

where f is the classifier and $l_i(\cdot,\cdot)$ are non-negative loss functions for each training sample. $r(\cdot)$ is a regularization term to avoid ill-posed problem as well as to prevent overfitting. Here L_2 regularization is adopted,

$$r(g) = \frac{1}{2}\sum_{m=1}^{M} \|\boldsymbol{w}_m\|_2^2 \tag{4.9}$$

The loss function should be chosen with more caution. Recall that the ultimate goal is to learn the acceptable set of a system. For most control problems in practice, the cost of assigning an unacceptable state as "good" (False Positive : $\{\hat{y} = 1 \mid y = -1\}$) is much higher than that of assigning an acceptable state as "bad" (False Negative : $\{\hat{y} = -1 \mid y = 1\}$). Hence in the learning framework, the False Negative Rate (FNR) and False Positive Rate (FPR) should be penalized differently. Assume that the rescaled cost of FNR and FPR are c_1 and c_2 respectively with $c_2 \geq c_1 \geq 1$, [1] the $0-1$ loss can be written as

$$L_{cs}^{0-1}(y, g) = \begin{cases} 0 & y = \text{sign}\{g\} \\ c_1 & y = 1 \text{ and } g < 0 \\ c_2 & y = -1 \text{ and } g > 0 \end{cases} \tag{4.10}$$

Seeing this, a naïve way to construct cost sensitive hinge loss, a convex approximation of (4.10), is to consider

$$L_{im}^{h}(y, g) = c_1 \mathbf{1}_{\{y=1\}} [1 - g]_+ + c_2 \mathbf{1}_{\{y=-1\}} [1 + g]_+ \tag{4.11}$$

where $[\cdot]_+ = \max\{0, \cdot\}$. Although the above loss is widely used in literature to formulate "cost sensitive" SVM, it is not Bayes consistent and the induced learning formulation only has limited capacity to enforce cost sensitivity especially when the difference between c_1 and c_2 is large. In this work we follow the lines of the work [148] and consider the following modified version of hinge loss

$$L_{cs}^{h}(y, g) = \mathbf{1}_{\{y=1\}} [1 - (2c_1 - 1)g]_+ + c_2 \mathbf{1}_{\{y=-1\}} [1 + g]_+$$

Proposition 21. *[148] $L_{cs}^{h}(y, g)$ is cost sensitive Bayes consistent, i.e. the associated cost sensitive risk is minimized by Bayes decision rule.*

[1] The cost rescaling will make risk analysis easier, and it is possible since for learning only the ratio of the costs matters

$$\begin{cases} g^*(\boldsymbol{x}) > 0 & \text{if } \mathbb{P}(y=1|\boldsymbol{x}) > \frac{c_1}{c_1+c_2} \\ g^*(\boldsymbol{x}) = 0 & \text{if } \mathbb{P}(y=1|\boldsymbol{x}) = \frac{c_1}{c_1+c_2} \\ g^*(\boldsymbol{x}) < 0 & \text{if } \mathbb{P}(y=1|\boldsymbol{x}) < \frac{c_1}{c_1+c_2} \end{cases} \tag{4.12}$$

Now by combining the regularization term and the loss function, we get the following cost sensitive large margin learning objective:

$$\begin{aligned} \min_{\boldsymbol{w}} \sum_{m=1}^{M} \frac{1}{2}||\boldsymbol{w}_m||^2 + \beta \sum_{i \in I^+} \Big[1 - (2c_1 - 1) \min_m \{\boldsymbol{w}_m^T \boldsymbol{x}_i\}\Big]_+ \\ + \beta \sum_{i \in I^-} \Big[c_2 + c_2 \min_m \{\boldsymbol{w}_m^T \boldsymbol{x}_i\}\Big]_+ \end{aligned} \tag{CPLM}$$

where β is the loss penalty hyperparameter. Similar to the soft margin C-SVM, it should be chosen with model selection techniques such as cross validation or solution path algorithms.

4.2 Learning a Structurally Imbalanced Classifier: Veto-classification

In this section, we propose Veto-Consensus Multiple Kernel Learning (VCMKL), a natural extension of CPLM but with hyperplanes in a transformed Hilbert space. VCMKL combines multiple kernels in a way that one class of samples is described by the logical intersection (consensus) of base kernelized decision rules, whereas the other classes by the union (veto) of their complements. The proposed configuration is a natural fit for domain description with multi-view learning, and can also be used for system fault detection, event diagnosis, etc. We first provide a generalization risk bound in terms of the Rademacher complexity of the classifier, and then formulate a large margin multi-ν learning objective with tunable training error bound.

As its name implies, VCMKL can be viewed as a version of multiple kernel learning (MKL). In recent years, MKL has shown promising results in a variety of applications and has attracted much attention in machine learning community. Given a set of base kernels, MKL finds an optimal combination of them with which an appropriate hypothesis is determined on the training data. A large body of literature has been addressing the arising issues of MKL, mainly from three perspectives and their intersections, i.e. theoretical learning bound, related optimization algorithm, and alternative MKL settings. To list a few, the generalization bounds for learning linear combination of multiple kernels have been extensively studied in [149, 150, 151] by analyzing various complexity metrics. Following the pioneer work [152] that formulates linear MKL as a semi-definite program (SDP), a series of work is devoted to improve the efficiency with various optimization techniques, such as reduced gradient [153], Newtown's method [154], and mirror descent [155]. Also data related issues such as sample adaptability and missing channels [156, 157] have been addressed. Despite the substantial

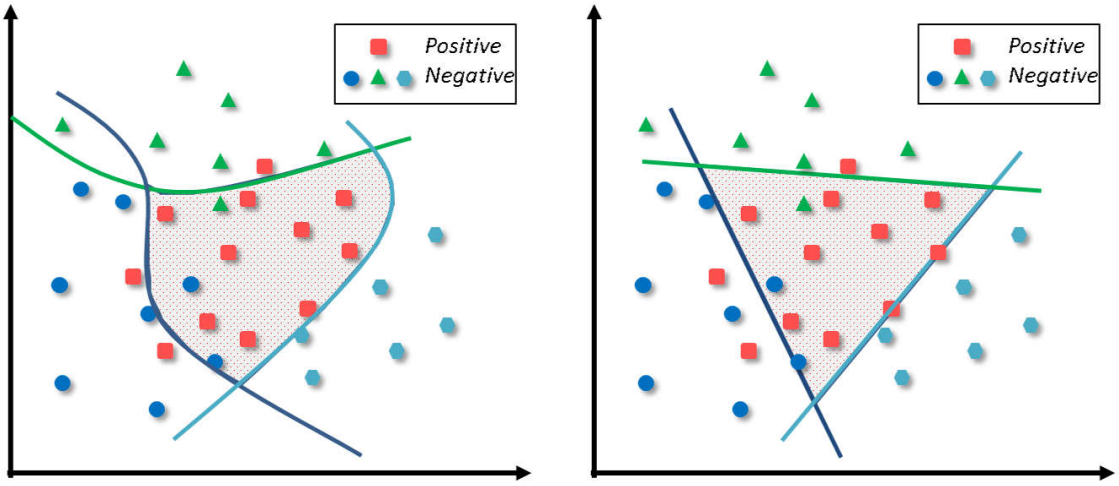


Figure 4.2: 2D VCMK with non-linear/all linear base kernels

theoretical advancement and algorithmic progress made in linear MKL, few of the results could be directly applied to MKL that incorporates nonlinear combinations. Indeed nonlinear MKL is usually studied on a case-by-case basis, such as polynomial combination [158], hierarchical kernel [159], hyperkernels [160], etc.

Now we explain the construction of VCMKL. To motivate the configuration, Figure 4.2 illustrates a practical problem where part of the classes contains hidden structures. In this example, the positive class is labeled. In contrast, the negative class contains several subgroups but only a"single" label is provided. To compensate for this implicit information, we propose to describe the positive class by the intersection of the acceptance region of multiple base kernel decision rules, and the negative class by the union of their complements. Hence a sample is classified as negative as long as one or more rules "votes" negative (Veto), and a positive assignment is made for a sample if and only if all of the rules agree (Consensus). With this intuition, VCMKL is a natural solution for applications involving hidden structures or multi-view features. Moreover because the construction inherently emphasizes the sensitivity to negative class and the specificity to positive class, it is also a promising tool for domain description problems.

In the sequel, we discuss the proposed VCMKL from both theoretical and algorithmic perspectives. Firstly, we formalize the the construction of the classifier and provide Rademacher complexity analysis. Then a large margin multi-ν learning formulation is proposed with training error controlled by hyperparameters.

4.2.1 The Classifier and Generalization Bound

Let $\mathcal{S} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{l}$ be a training data set, where $\boldsymbol{x} \in \mathbb{R}^d$ and d is the dimension of features. Without loss of generality let $y \in \{+1, -1\}$ indicate class labels, with negative class contains hidden subgroups. Consider a feature mapping $\phi : \mathbb{R}^d \to \mathcal{H}$, in the new Hilbert space a

hyperplane can be written as $f(\boldsymbol{x}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle_{\mathcal{H}} + b$. A decision rule for data $\boldsymbol{x}'$ is given by the sign of $f(\boldsymbol{x}')$. Formalizing the idea of using intersections of M base kernel mappings for positive class and the union of their complement for negative class, the composed classifier $g(\cdot)$ is similar to CPML

$$\begin{aligned} \{g(\boldsymbol{x}) > 0\} &= \{f_1(\boldsymbol{x}) > 0\} \cap \cdots \cap \{f_M(\boldsymbol{x}) > 0\} \\ &= \left\{ \min_{1,\cdots,M} \{f_1(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x})\} > 0 \right\} \end{aligned}$$

On the other hand, the acceptance region for negative class is

$$\left\{ \min_{1,\cdots,M} \{f_1(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x})\} \leq 0 \right\}.$$

For short notation, let us denote $\langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle_{\mathcal{H}} = \boldsymbol{w} \cdot \phi(\boldsymbol{x})$ as the inner product and $||\cdot||$ as the corresponding norm in $\mathcal{H}$. Then the combined classifier is simply

$$g(\boldsymbol{x}) = \min\{\boldsymbol{w}_1 \cdot \phi(\boldsymbol{x}) + b_1, \cdots, \boldsymbol{w}_M \cdot \phi(\boldsymbol{x}) + b_M\}$$

Note the similarity of the VCMKL construction with the CPLM proposed in last section. With all linear base kernels, the VCMKL essentially reduces to the convex piece-wise linear classifier.

Before proceeding to any method to learn this classifier, we conduct complexity analysis in MKL framework for generalization bound and model selection purpose. Let the function class of g be denoted as $\mathcal{G}$, and that of f_j be denoted as $\mathcal{F}_j$. As a classic measure of richness, the *Empirical Rademacher Complexity* for a function class $\mathcal{F}$ is defined as $\widehat{\mathcal{R}}(\mathcal{F}(\boldsymbol{x}_1^l)) \triangleq E_{\boldsymbol{\sigma}} \left[\sup_{f \in \mathcal{F}} \left| \frac{2}{l} \sum_{i=1}^{l} \sigma_i f(\boldsymbol{x}_i) \right| \right]$ where $\sigma_1, \cdots, \sigma_l$ are i.i.d. Rademacher variables such that $\mathbb{P}(\sigma_i = +1) = \mathbb{P}(\sigma_i = -1) = 1/2$. The definition measures the complexity/richness of function class $\mathcal{F}$ in terms of its ability to "match" Rademancher variables. With Talagrand's Lemma and an induction argument, we show that

Theorem 22. *The function class $\mathcal{G}$ of VCMKL has*

$$\widehat{\mathcal{R}}(\mathcal{G}\left(\boldsymbol{x}_1^l\right)) \leq 2 \sum_{j=1}^{M} \widehat{\mathcal{R}}\left(\mathcal{F}_j(\boldsymbol{x}_1^l)\right)$$

Further assume $\mathcal{F}_j$ forms a bounded function class with kernel $\kappa_j(\cdot,\cdot)$ and kernel matrix $\boldsymbol{K}_j$ such that $\mathcal{F}_j = \left\{ \boldsymbol{x} \mapsto \sum_{i=1}^{l} \alpha_i \kappa_j(\boldsymbol{x}_i, \boldsymbol{x}) \mid \boldsymbol{\alpha}^T \boldsymbol{K}_j \boldsymbol{\alpha} \leq B_j \right\}$ then

$$\widehat{\mathcal{R}}\left(\mathcal{G}(\boldsymbol{x}_1^l)\right) \leq \frac{4}{l} \sum_{j=1}^{M} B_j \sqrt{tr(\boldsymbol{K}_j)}.$$

In general it is hard to tighten the additive nature of the complexity. With the above results at hand, the generalization guarantee of the MKVCL can be obtained immediately from the classic results in the PAC learning framework. Let $L(g) = E_{\mathcal{S}}[\mathbf{1}_{\text{sgn}(g(\boldsymbol{x}))\neq y}]$ be the generalization error of the MKVC classifier g, and let $\widehat{L}_\rho(g) \triangleq \frac{1}{l}\sum_{i=1}^{l} \Psi_\rho\left(y_i g(\boldsymbol{x}_i)\right)$ be the empirical $\rho - margin$ loss with $\Psi_\rho(t) = [\min\{1, 1 - t/\rho\}]_+$. Then we have with probability at least $1-\delta$

$$L(g) \leq \widehat{L}_\rho(g) + \frac{8}{\rho}\sum_{j=1}^{M} \frac{B_j\sqrt{\text{tr}(\boldsymbol{K}_j)}}{l} + 3\sqrt{\frac{\log(2/\delta)}{2l}}$$

4.2.2 Multi-ν learning Learning Objective

To learn the multi-kernels classifier from data, we adopt a learning objective that maximizes the total margin defined in [146] while minimizing the hinge loss of misclassifications. Inspired by the advantages of νSVM [161], the following multi-ν learning formulation is proposed:

$$\begin{aligned}\min_{\boldsymbol{w}_m, b_m, \rho_m} & \frac{1}{2}\sum_{m=1}^{M}||\boldsymbol{w}_m||^2 - \sum_{m=1}^{M}\nu_m\rho_m \\ & + \frac{\gamma}{l}\sum_{i\in I^+}\max_m\left\{[\rho_m - y_i(\boldsymbol{w}_m\cdot\boldsymbol{x}_i + b_m)]_+\right\} \\ & + \frac{1-\gamma}{l}\sum_{i\in I^-}\min_m\left\{[\rho_m - y_i(\boldsymbol{w}_m\cdot\boldsymbol{x}_i + b_m)]_+\right\}\end{aligned} \tag{\textbf{VCMKL}}$$

where I^+ and I^- are the index sets of positive and negative samples, respectively. The hyperparameters $\nu_1, \cdots, \nu_M \in [0,1]$ weigh the margins, and two types of losses are treated differently by introducing $\gamma \in [0,1]$. The multi-ν formulation still reflects the Veto-Consensus intuition: the loss for positive class is the maximum over all decision boundaries, while for negative class only the one with minimum loss is counted. The first three terms in the above minimization problem are convex, while the last term is non-convex as the minimum of M truncated hyperplanes. To obtain an equivalent dual form of the learning objective, we start by considering a weighted version of the M losses over a simplex:

$$L_{avg}(\boldsymbol{w}, \boldsymbol{x}_i, \boldsymbol{\lambda}_i) = \sum_{m=1}^{M}\lambda_{im}[\rho_m - y_i(\boldsymbol{w}_m\cdot\boldsymbol{x}_i + b_m)]_+ \tag{4.13}$$

with $\boldsymbol{\lambda}_i \in \left\{\boldsymbol{\lambda}_i : \sum_{m=1}^{M}\lambda_{im} = 1, \lambda_{im} \geq 0\right\} \triangleq \mathbb{S}^M$, a row vector in the $|I^-| \times M$ matrix $\boldsymbol{\lambda}$ containing the loss weighting parameters of negative samples. Denote $L_{min}(\boldsymbol{w}, \boldsymbol{x}_i) = \min_m\left\{[\rho_m - y_i(\boldsymbol{w}_m\cdot\boldsymbol{x}_i + b_m)]_+\right\}$ as the original loss for $\boldsymbol{x}_i$, it is straightforward that

$$L_{min}(\boldsymbol{w}, \boldsymbol{x}_i) = \min_{\boldsymbol{\lambda}_i\in\mathbb{S}^M} L_{avg}(\boldsymbol{w}, \boldsymbol{x}_i, \boldsymbol{\lambda}_i) \tag{4.14}$$

With this trick we reformulate the learning objective as

Proposition 23. *(**VCMKL**) is equivalent to*

$$\begin{aligned}\min_{\boldsymbol{\lambda}_i\in\mathbb{S}^M}\min_{\substack{\boldsymbol{w}_m,b_m,\\ \rho_m\geq 0}}\ & \frac{1}{2}\sum_{m=1}^{M}||\boldsymbol{w}_m||^2-\sum_{m=1}^{M}\nu_m\rho_m \\ & +\frac{\gamma}{l}\sum_{i\in I^+}\max_m\left\{[\rho_m-y_i(\boldsymbol{w}_m\cdot\boldsymbol{x}_i+b_m)]_+\right\} \\ & +\frac{1-\gamma}{l}\sum_{i\in I^-}\sum_{m=1}^{M}\lambda_{im}[\rho_m-y_i(\boldsymbol{w}_m\cdot\boldsymbol{x}_i+b_m)]_+\end{aligned} \tag{\textbf{Primal}}$$

The newly introduced variables $\boldsymbol{\lambda}$ can be viewed as hidden subgroup indicators, hence VCMKL can indeed be thought of as a multi-kernel extension of learning with latent variables. Considering the form of the Primal, it is tempting to apply CCCP and alternating heuristics. Yet in this work a rigorous optimization algorithm will be developed to approach global optimum. But before that let's look into the relation between training error and the hyperparameters ν_m, γ in the learning formulation. Replacing the inner optimization of the Primal with its dual, we obtain that the Primal is equivalent to

$$\begin{aligned}&\min_{\boldsymbol{\lambda}_i\in\mathbb{S}^M}\mathcal{J}_d(\boldsymbol{\lambda}) \quad \text{where} \\ &\mathcal{J}_d(\boldsymbol{\lambda})=\begin{cases}\min_{\boldsymbol{\alpha}}\frac{1}{2}\sum_{m=1}^{M}\sum_{i,j=1}^{l}\alpha_{im}y_i\kappa_m(\boldsymbol{x}_i,\boldsymbol{x}_j)y_j\alpha_{jm} \\ \textit{subject to} \\ \alpha_{im}\geq 0 \quad \forall i,\ \forall m \\ \alpha_{im}\leq\frac{1-\gamma}{l}\lambda_{im} \quad \forall i\in I^-,\ \forall m \\ \sum_{m=1}^{M}\alpha_{im}\leq\frac{\gamma}{l} \quad \forall i\in I^+ \\ \sum_{i=1}^{l}\alpha_{im}\geq\nu_m \quad \forall m \\ \sum_{i=1}^{l}\alpha_{im}y_i=0 \quad \forall m\end{cases}\end{aligned} \tag{\textbf{Dual}}$$

To see the effect of hyperparameters, it is useful to define partition of samples similar to the classical SVM:

Definition 7. *Partition of Samples : Based on the value of α_{im} at optimal, the i^{th} sample is called*

- *positive support vector if $i\in I^+$ and $\sum_m\alpha_{im}>0$.*
- *positive bounded support vector if $i\in I^+$ and $\sum_m\alpha_{im}=\frac{\gamma}{l}$.*
- *negative support vector of class m if $i\in I^-$ and $\alpha_{im}>0$.*
- *negative bounded support vector of class m if $i\in I^-$ and $\alpha_{im}=\frac{1-\gamma}{l}$.*

All the other samples are called non-support vectors. The following proposition relates the choice of hyperparameters to the training error tolerance.

Proposition 24. *Define* $\nu^+ = \frac{l\sum_m \nu_m}{2\gamma|I^+|}$ *and* $\nu_m^- = \frac{l\nu_m}{2(1-\gamma)|I^-|}$*, and denote* $N^{sv+}, N_m^{sv-}, N^{bsv+}, N_m^{bsv-}$ *as the number of all postive/negative support vectors, positive/negative bounded support vectors, respectively, then*

$$\frac{N^{bsv+}}{|I^+|} \leq \nu^+ \leq \frac{N^{sv+}}{|I^+|} \tag{4.15a}$$

$$\frac{N_m^{bsv-}}{|I^-|} \leq \nu_m^- \leq \frac{N_m^{sv-}}{|I^-|} \tag{4.15b}$$

Form the right hand side, ν^+ and ν_m^- give a lower bound on the fraction of positive support vectors and negative support vectors of class m, respectively. The left hand side upper bound is more interesting: by definition, $N^{bsv+}/|I^+|$ and $N_m^{bsv-}/|I^-|$ are respectively the training false negative error and false positive error of class m. Hence the bound implies that one can impose smaller training error of different types by decreasing the corresponding ν. The role of γ is also significant: it can incorporate an uneven consideration of training errors committed in two classes, which can be harnessed to handle imbalanced availability of positive/negative samples. In short, the advantage of the multi-ν formulation is that the training result can be controlled simply by tuning bounds as a function of hyperparameters.

4.2.3 Extension to Hidden Structrued Semi-supervised Machine (HS^{3}M)

Traditionally, there have been two fundamentally different paradigms of machine learning (ML). The first one is supervised learning, with the goal of learning a mapping from some input $\boldsymbol{x}$ to output y. Usually the observations $(\boldsymbol{x}_i, y_i), i = 1, \cdots, n$ are called samples, $\boldsymbol{x}_i$ are referred to as features of sample i, and $y_i \in \mathcal{Y}$ are called labels or targets. To find the "optimal" mapping f, the learning task can be formulated into for example "regularized empirical risk minimization". The second task of ML is unsupervised learning. Under this setting, only the unlabeled observations $X = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$ are given. Typically, the goal of unsupervised learning is to identify interesting structures in the data X, such as clusters, quantiles, support, low-dimensional embedding, or more generally the patterns related to the distribution of the data.

The presence of both labeled and unlabeled data motivates the so-called semi-supervised learning [162, 163, 164]. The hope is that, by combining both types of available data sets, semi-supervised learning could find better models/classifiers, and reduce the cost of expert engagement. In the context of machine learning for cyber physical systems, data with detailed labels is precious but scant. On the other hand partly labeled data with incomplete information may be obtained with less cost or by using decision support systems. Moreover unlabeled data is usually available in large quantities simply by collecting measurements of the system. To formalize the information availability in different scenarios, consider data in the following three formats:

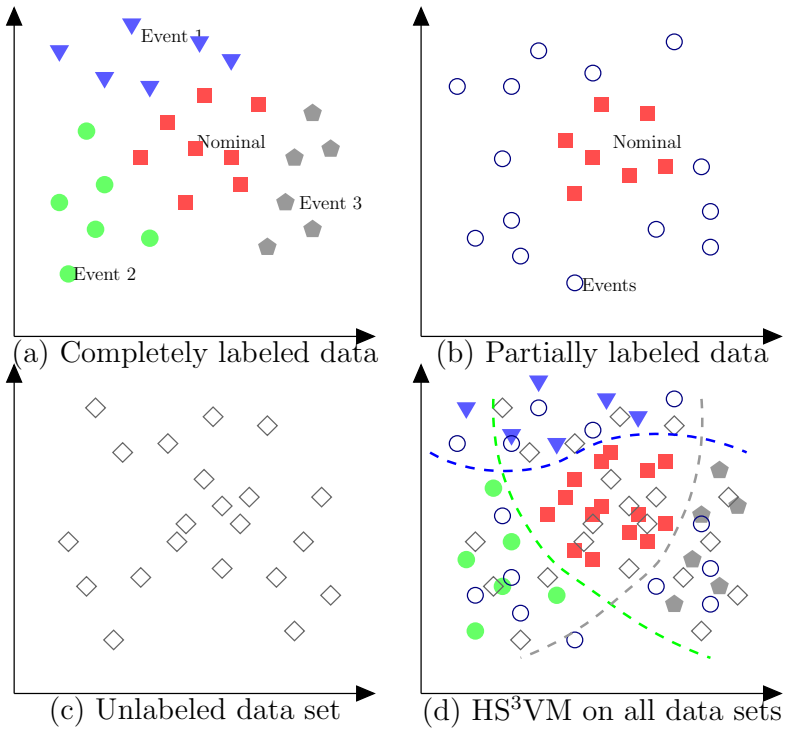
1 Completely labeled data samples, denoted as $\{\boldsymbol{x}_i, y_i, z_i\}$, where i is the sample index, $\boldsymbol{x}_i$ is the system measurement, $y_i \in \{+1, -1\}$ is a "cursory label", and if $y_i = +1$, a "detailed label" $z_i \in \{1, \cdots, K\}$ is provided.

2 Partly labeled data samples, denoted as $\{\boldsymbol{x}_i, y_i, \cdot\}$, where y_i is still the cursory label, but when $y_i = -1$, no other information is provided. We refer this case as partial labeling or label with hidden subgroups.

3 Unlabeled data samples, denoted as $\{\boldsymbol{x}_i, \cdot, \cdot\}$, where only features $\boldsymbol{x}_i$ is accessible.

The above data type division is better understood in the application of event classification (or called system diagnosis): the cursory label is an indicator for "nominal/stable state" ($y = +1$) or an interesting "event" ($y = -1$). If $y = +1$, an expert can be inquired to provide an event type $z_i \in \{1, \cdots, K\}$ (Type 1 data), or in the case of missing expert input, only the cursorily labeled data is recorded (Type 2 data). Finally, unlabeled data can be accumulated simply from sensor measurement of the system.

An illustration of these different situations is given in Figure 4.3. Intuitively, the partly labeled data should be helpful: at least it provides discriminating information for a binary classification. The role of unlabeled data might be ambiguous since it does not carry any expert knowledge. However, it does contain distributional information of measurement, which could be exploited with a proper formulation. As an effort to combine all three information sources, we propose a unified learning objective that makes the best use of partial knowledge to improve classification performance.



(a) Completely labeled data (b) Partially labeled data

(c) Unlabeled data set (d) HS^{3}VM on all data sets

Figure 4.3: Different data format and the intuition of HS^{3}M

Comparing Figure 4.3(a) and 4.3(b), we see that partly labeled data could be viewed as data with "missing detailed labels" in the positive class (events types). To compensate for this implicit information, the VCMKL constructed in previous section is a nature fit: one can describe the negative class (stable state) by the intersection of the acceptance region of multiple base kernel decision rules, while the positive class (events class) by the union of their complements, as is shown in Figure 4.3(d). Consider a feature mapping $\phi : \mathbb{R}^d \to \mathcal{H}$, in the new Hilbert space we write a hyperplane classifier as $f(\boldsymbol{w}, \boldsymbol{x}) = \langle \boldsymbol{w}, \phi(\boldsymbol{x}) \rangle_{\mathcal{H}} + b \triangleq \boldsymbol{w} \cdot \phi(\boldsymbol{x})$ for short hand notation. Then with VCMKL the proposed classifier is

$$g(\boldsymbol{w}, \boldsymbol{x}) = \min_k \{\boldsymbol{w} \cdot \phi_k(\boldsymbol{x})\} \tag{4.16}$$

and the corresponding hinge loss for a partly labeled data sample $\{\boldsymbol{x}_i, y_i, \cdot\}$ is just

$$\left[1 - y_i \min_k \{\boldsymbol{w} \cdot \phi_k(\boldsymbol{x_i})\}\right]_+ \tag{4.17}$$

Having composed the classifier, we adopt a tentative labeling strategy to include information provided by unlabeled data. More specifically we consider

$$\widehat{y}_i = \text{sign}\left(\min_k \{\boldsymbol{w} \cdot \phi_k(\boldsymbol{x}_i)\}\right) \tag{4.18}$$

then the corresponding hinge loss has the form

$$\left[1 - \widehat{y}_i \min_k \{\boldsymbol{w} \cdot \phi_k(\boldsymbol{x_i})\}\right]_+ = \left[1 - \left|\min_k \{\boldsymbol{w} \cdot \phi_k(\boldsymbol{x}_i)\}\right|\right]_+$$

Putting things together, we propose the following regularized hinge loss minimization for event detection that incorporates all explicit and partial expert knowledge:

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \frac{1}{2}||\boldsymbol{w}||^2_{\mathcal{H}} + c_1 \sum_{i \in \mathcal{L}^+} \left[1 - \min_k \{\boldsymbol{w} \cdot \phi_k(\boldsymbol{x_i})\}\right]_+ \\ & + c_{21} \sum_{i \in \mathcal{L}_1^-} [1 + \boldsymbol{w} \cdot \phi_{z_i}(\boldsymbol{x}_i)]_+ \\ & + c_{22} \sum_{i \in \mathcal{L}_2^-} \left[1 + \min_k \{\boldsymbol{w} \cdot \phi_k(\boldsymbol{x}_i)\}\right]_+ \\ & + c_3 \sum_{i \in \mathcal{U}} \left[1 - \left|\min_k \{\boldsymbol{w} \cdot \phi_k(\boldsymbol{x}_i)\}\right|\right]_+ \end{aligned} \tag{\textbf{HS3M}}$$

where we have denoted $\mathcal{L}^+$ as the index set of all data samples that has $y_i = +1$, including both completely and partly labeled samples, $\mathcal{L}_1^-$ as the index set of completely labeled samples with $y_i = -1$ and event type z_i (hence the hinge loss only involves the corresponding individual classifier f_{z_i}). The index set $\mathcal{L}_2^-$ contains partly labeled samples in the event class, and $\mathcal{U}$ is the index of all unlabeled data samples. The loss penalty hyper-parameters c_1-c_3 weigh each loss term differently, and should be chosen by taking into account the imbalanced cost for false positive and false negative error, sample size in each category, as well as for model selection considerations. Since the above formulation deals with both hidden structures and unlabeled samples in the available data, we call it Hidden Structured Semi-Supervised Machine (HS^{3}M).

To enable the usage of kernel trick in the dual form, we transform **HS3M** by introducing additional "hidden" variables, and similarly to the reformulation of VCMKL, we can write the learning objective in the following joint optimization form:

Proposition 25. ***HS3M*** *is equivalent to*

$$\begin{aligned}\min_{\boldsymbol{\eta},\boldsymbol{\zeta}}\min_{\boldsymbol{w}} \quad & \frac{1}{2}||\boldsymbol{w}||_{\mathcal{H}}^2 + c_1\sum_{i\in\mathcal{L}^+}\Big[1-\min_k\{\boldsymbol{w}\cdot\phi_k(\boldsymbol{x_i})\}\Big]_+ \\ & + c_{21}\sum_{i\in\mathcal{L}^-}\left[1+\boldsymbol{w}\cdot\phi_{z_i}(\boldsymbol{x}_i)\right]_+ \\ & + c_{22}\sum_{i\in\mathcal{L}_H^-}\sum_{k=1}^{K}\eta_{ik}\left[1+\boldsymbol{w}\cdot\phi_k(\boldsymbol{x}_i)\right]_+ \\ & + c_3\sum_{i\in\mathcal{U}}\sum_{k=1}^{K}\zeta_{ik}\left[1+\boldsymbol{w}\cdot\phi_k(\boldsymbol{x}_i)\right]_+ \\ & + c_3\sum_{i\in\mathcal{U}}\zeta_{i(K+1)}\max_j\{0,1-\boldsymbol{w}\cdot\phi_j(\boldsymbol{x}_i)\} \\ \textit{subject to} \quad & \boldsymbol{\eta}_i\in\mathbb{S}^K,\ \forall i\in\mathcal{L}_H^-;\quad \boldsymbol{\zeta}_i\in\mathbb{S}^{K+1},\ \forall i\in\mathcal{U}\end{aligned}$$

In addition, the two minimizations are interchangeable.

The introduced variables η and ζ can be thought of as hidden state indicators for partially labeled data and unlabeled data, respectively. The corresponding dual for the inner optimization is

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}}\frac{1}{2}\left\|\sum_{k,i\in\mathcal{I}}\alpha_{ik}y_i\phi_k(\boldsymbol{x}_i)+\sum_{i\in\mathcal{L}^-}\beta_i\phi_{z_i}(\boldsymbol{x}_i)+\sum_{k,i\in\mathcal{U}^+}\gamma_{ik}\phi_k(\boldsymbol{x}_i)\right\|_{\mathcal{H}}^2-\sum_{k,i\in\mathcal{I}}\alpha_{ik}-\sum_{i\in\mathcal{L}^-}\beta_i-\sum_{k,i\in\mathcal{U}^+}\gamma_{ik}$$

subject to

$$\begin{cases}\alpha_{ik}\geq 0;\quad \sum_k\alpha_{ik}\leq c_1 \quad \forall i\in\mathcal{L}^+\\ 0\leq\beta_i\leq c_{21}\quad\forall i\in\mathcal{L}^-\\ 0\leq\alpha_{ik}\leq c_{22}\eta_{ik}\quad\forall i\in\mathcal{L}_H^-\\ 0\leq\alpha_{ik}\leq c_3\zeta_{ik}\quad\forall i\in\mathcal{U}^-\\ \gamma_{ik}\geq 0;\quad\sum_k\gamma_{ik}\leq c_3\zeta_{i(K+1)}\quad\forall i\in\mathcal{U}^+\\ \sum_{k,i\in\mathcal{I}}y_i\alpha_{ik}+\sum_{k,i\in\mathcal{L}^-}y_i\beta_i+\sum_{k,i\in\mathcal{U}^+}y_i\gamma_{ik}=0\end{cases}$$

(Inner Dual)

where the Lagrangian multipliers $\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}$ for the inner optimization of the primal are now decision variables. Note that the unlabeled data set $\mathcal{U}$ is used as two dummy copies with tentative labels $y_i=+1$ for $i\in\mathcal{U}^+$ and $y_i=-1$ for $i\in\mathcal{U}^-$, respectively. Also for short hand notation, define $\mathcal{I}\triangleq\mathcal{L}^+\cup\mathcal{L}_H^-\cup\mathcal{U}^-$, and a unified decision variable

$$\boldsymbol{\theta}=\left[\boldsymbol{\alpha}^T,\boldsymbol{\beta}^T,\boldsymbol{\gamma}^T\right]^T$$

where we write

$$\begin{aligned}\boldsymbol{\alpha} &\triangleq \left[\alpha_{11}, \cdots, \alpha_{|\mathcal{I}|1}, \alpha_{12} \cdots, \alpha_{|\mathcal{I}|K}\right]^T \\ \boldsymbol{\beta} &\triangleq \left[\beta_1, \cdots, \beta_{|\mathcal{L}^-|}\right]^T \\ \boldsymbol{\gamma} &\triangleq \left[\gamma_{11}, \cdots, \gamma_{|\mathcal{I}|1}, \gamma_{12} \cdots, \gamma_{|\mathcal{I}|K}\right]^T\end{aligned} \tag{4.19}$$

It is immediate that the norm in the Hilbert space reduces to inner products. Hence the objective of the dual can be equivalently written as

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \sum_{i,j} \theta_i \langle \phi(\boldsymbol{x}_i, y_i, i), \phi(\boldsymbol{x}_j, y_j, j) \rangle \theta_j - \sum_i \theta_i \tag{4.20}$$

in which the so called kernel trick could be used for direct computation of the inner product without the need to compute the explicit feature mapping $\phi(\cdot)$, i.e.

$$\langle \phi(\boldsymbol{x}, y, i), \phi(\boldsymbol{x}', y', j) \rangle = \kappa_{(iy)(jy')}(\boldsymbol{x}, \boldsymbol{x}') \tag{4.21}$$

For a more compact form of the dual, let us further define a matrix $\boldsymbol{Q}$ with elements $\boldsymbol{Q}_{(iy)(jy')} = \kappa_{(iy)(jy')}(\boldsymbol{x}, \boldsymbol{x}')$, a column vector of all hidden variables

$$\boldsymbol{\lambda} \triangleq \{\boldsymbol{\eta}_1; \cdots; \boldsymbol{\eta}_{|\mathcal{L}_H^-|}, \boldsymbol{\zeta}_1, \cdots, \boldsymbol{\zeta}_{|\mathcal{U}|}\},$$

and a augmented label vector (with dummy copies of unlabeled set) as

$$\widetilde{\boldsymbol{y}} = [\underbrace{1, \cdots, 1}_{\mathcal{L}^+}, \underbrace{-1, \cdots, -1}_{\mathcal{L}_H^-}, \underbrace{-1, \cdots, -1}_{\mathcal{U}^-}, \underbrace{-1, \cdots, -1}_{\mathcal{L}^-}, \underbrace{1, \cdots, 1}_{\mathcal{U}^+}]^T$$

together with a matrix encapsulated inequality constraints, the (negative) inner optimization becomes

$$\begin{aligned}&\min_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^T \boldsymbol{Q} \boldsymbol{\theta} - \boldsymbol{e}^T \boldsymbol{\theta} \\ &\text{subject to} \begin{cases} \boldsymbol{C}^\theta \boldsymbol{\theta} \leq \boldsymbol{C}^\lambda \boldsymbol{\lambda} + \boldsymbol{C}^0 \\ \widetilde{\boldsymbol{y}}^T \boldsymbol{\theta} = 0. \end{cases}\end{aligned} \tag{OPT_D}$$

where $\boldsymbol{C}^\theta, \boldsymbol{C}^\lambda, \boldsymbol{C}^0$ are constant matrices with $K|\mathcal{L}_H^-| + (K+1)|\mathcal{U}|$ rows. Similar to other kernel methods in machine learning, HS^{3}M is restricted to Mercer kernels, thence $\boldsymbol{Q}$ is positive definite, and (Dual) is in a convex quadratic program. The learning objective **HS3M** now becomes

$$\max_{\boldsymbol{\lambda} \in \Lambda} \min_{\theta \in \Theta(\boldsymbol{\lambda})} \mathcal{J}(\boldsymbol{\theta}) \tag{4.22}$$

4.3 Derivation and Applications of the PDM Algorithm

In this section we propose a novel optimization algorithm to solve the machine learning problems formulated in last section. More broadly, we consider a class of non-convex learning

problems that can be formulated into jointly optimizing regularized hinge loss and a set of auxiliary variables. Besides the three models proposed in last section, such problems encompass but are not limited to various versions of semi-supervised learning, learning with hidden structures, robust learning, etc. Existing methods either suffer from local minima or have to invoke a non-scalable combinatorial search. In this section, we propose a learning procedure (partly based on our work [165]), namely Parametric Dual Maximization (PDM), that can approach global optimality efficiently with user specified approximation levels . The building blocks of PDM are two new results: (1) The equivalent convex maximization reformulation derived by parametric analysis. (2) The improvement of local solutions based on a necessary and sufficient condition for global optimality. Since PDM is not limited to learning problems considered in this chapter, but applies to a much broader group of non-Convex machine learning, we discuss PDM in a general ML framework and adopt self-contained notations in this section.

4.3.1 Related Works

To enhance the performance on more challenging tasks, variations of the classic large margin learning formulation are proposed to incorporate additional modeling flexibility. To name a few, semi-supervised SVM (S^3VM) is introduced in [166, 167] to combine labeled and unlabeled samples together for overall risk minimization. To learn a classifier for datasets having unobserved information, SVM with latent variables is proposed in [168] for object detection and in [169, 170] for structural learning. Inasmuch as the traditional large margin classifier with hinge loss can be sensitive to outliers, the authors of [171] suggest a ramp loss with which a robust version of SVM is proposed.

Nonetheless, unlike the classical SVM learning objective that possesses amiable convexity, these variations introduce non-convex learning objectives, hindering their generalization performance and scalable deployment due to optimization difficulties. In literature, much effort has been made to obtain at least a locally optimal solution: Viewing the problem as a biconvex optimization leads to a series of alternating optimization (AO) algorithms. For example, in [168], latent SVM was trained by alternately solving standard SVM and updating latent variables. Another widely applied technique is the concave-convex Procedure (CCCP) [172]. Among many others, [169, 173] used CCCP for latent structural SVM training. Direct application of the gradient-based method is especially attractive for large scale problems owing to its low computational cost [174]. Such examples include the stochastic gradient descent (SGD) for large margin polytope machine [146, 175] and S^3VM [176]. Combinatorial optimization methods, e.g., the local search method [167] and branch and bound (B & B) [177], were also implemented for small-scale problems. It's worth mentioning that other heuristic approaches and relaxations such as continuation method [178] and semidefinite program (SDP) relaxation [179, 171] have also been examined for several applications.

Yet except B & B, all of the aforementioned methods, i.e., AO, CCCP, and SGD, only converge to local minimums and could be very sensitive to initial conditions. Although SDP approximation yields a convex problem, the quality of the relaxation is still an open question

in both theory and practice [180]. On the other hand, it has long been realized that global optimal solution can return excellent generalization performance in situations where local optimal solutions fail completely [177]. The major issue with B & B is its scalability: the size of the search tree can grow exponentially with the number of integer variables [181], making it only suitable for small scale problems. Interested readers are referred to [182] for a thorough discussion.

In this work, we propose a learning procedure, namely Parametric Dual Maximization (PDM), based on a different view of the problem. We first demonstrate that the learning objectives can be rewritten into jointly optimizing regularized hinge loss and a set of auxiliary variables. Then we show that they are equivalent to non-smooth convex maximization through a series of parametric analysis techniques. Finally, we establish PDM by exploiting a necessary and sufficient global optimality condition. Our contributions are highlighted as follows. (1) The equivalence to non-smooth convex maximization unveils a novel view of an important class of learning problems such as S^{3}VM. Now we know that they are *NP-hard*, but possesses gentle geometric properties that allow new solution techniques. (2) We develop a set of new parametric analysis techniques, which can be reused for many other tasks, e.g., solution path calculation. (3) By checking a necessary and sufficient optimality condition, the proposed PDM can approach the global optimum efficiently with user specified approximation levels.

4.3.2 A Class of Non-Convex Learning Problems

A labeled data sample is denoted as $(\boldsymbol{x}_i, y_i)$, with $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$. We focus on the following joint minimization problem

$$\min_{\boldsymbol{p}\in\mathbb{P}} \min_{\boldsymbol{w},b} \quad \mathcal{P}(\boldsymbol{w}, b; \boldsymbol{p}) = \frac{1}{2}||\boldsymbol{w}||_{\mathcal{H}}^2 + \sum_{i=1}^{N} c_i p_i V(y_i, h_i) \tag{OPT1}$$

where $h_i = \kappa(\boldsymbol{w}, \boldsymbol{x}_i) + b$ with $\kappa(\cdot, \cdot)$ a Mercer kernel function. The function V is the Hinge loss, i.e., $V(y_i, h_i) = \max(0, 1 - y_i h_i)$. We call $\boldsymbol{p} \triangleq [p_1, \cdots, p_N]^T \in \mathbb{P}$ the auxiliary variable of the problem, and assume its feasible set $\mathbb{P}$ to be convex. note that with $\boldsymbol{p}$ fixed, the inner problem resembles traditional large margin learning. Depending on the context, the auxiliary variable $\boldsymbol{p}$ can be regarded as hidden states or probability assignments for loss terms. We focus on (OPT1) in this work, because the three learning problems considered in this chapter, as well as many other large margin learning variations, including S^{3}VM, latent SVM, robust SVM, etc., can be rewritten in this form. The following is another example of such reformulation:

Example 1. *Consider the learning objective of Semi Supervised Support Vector Machine (S^3 VM):*

$$\min_{\boldsymbol{w},b,\widehat{\boldsymbol{y}}_u} \frac{1}{2}||\boldsymbol{w}||_{\mathcal{H}}^2 + C_1 \sum_{i=1}^{l} V(y_i, h_i) + C_2 \sum_{i=l+1}^{n} V(\widehat{y}_i, h_i)$$

where l is the number of labeled samples and $n-l$ unlabeled samples are included in the loss with "tentative" label $\widehat{\boldsymbol{y}}_u$, which constitute additional variables to minimize over. Interestingly, the learning objective has the following equivalent form:

$$\min_{\boldsymbol{w},b} \min_{\boldsymbol{p}} \frac{1}{2}||\boldsymbol{w}||_{\mathcal{H}}^2 + C_1 \sum_{i=1}^{l} V(y_i, h_i) + C_2 \sum_{i=l+1}^{n} [p_i V(1, h_i) + (1-p_i)V(-1, h_i)]$$

The equivalence is due to the fact that minimizing over p_i will cause all its mass to concentrate on the smaller of $V(1, h_i)$ and $V(-1, h_i)$. Formally for any variables $\xi_1, \cdots, \xi_M$ we have $\min_m\{\xi_1, \cdots, \xi_M\} = \min_{\boldsymbol{p}\in\mathbb{S}^M} \sum_{m=1}^{M} p_m \xi_m$, where $\mathbb{S}^M$ is the simplex in $\mathbb{R}^M$. Due to strict feasibility and biconvexity in $(\boldsymbol{w}, b)$ and $\boldsymbol{p}$, we can exchange the order of minimization and obtain an equivalent form similar to (OPT1). The variable p_i is the "probability" of $\widehat{y}_i = 1$.

Observing that the inner problem of OPT1 is convex quadratic with fixed $\boldsymbol{p}$, we replace it with its dual problem and cast OPT1 into

$$\begin{aligned} &\max_{\boldsymbol{p}\in\mathbb{P}} \min_{\boldsymbol{\alpha}\in\mathbb{A}(\boldsymbol{p})} \mathcal{J}(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i,j} \alpha_i y_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) y_j \alpha_j - \sum_i \alpha_i \\ &\text{where } \mathbb{A}(\boldsymbol{p}) = \left\{\boldsymbol{\alpha} \mid 0 \le \alpha_i \le c_i p_i \ \forall i,\ \boldsymbol{y}^T\boldsymbol{\alpha} = 0\right\} \end{aligned} \tag{OPT2}$$

In the above equivalent formulation, we can view the inner optimization as minimizing a quadratic function subject to polyhedron constraints that are *parametrized* by the auxiliary variable $\boldsymbol{p}$. Assuming the kernel matrix $\boldsymbol{K}$, defined by $\boldsymbol{K}_{i,j} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$, is strictly positive[2], then the optimum is unique by strict convexity, and the solution $\boldsymbol{\alpha}^*$ is a function of $\boldsymbol{p}$. Ideally, if one can write out the functional dependence explicitly, OPT2 is essentially $\max_{\boldsymbol{p}\in\mathbb{P}} \mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{p}))$, which minimizes over the "parameters" $\boldsymbol{p}$ of the inner problem. In the terminology of operational research and optimization, the task of analyzing the dependence of an optimal solution on multiple parameters is called parametric programming. Inspired by this new view of OPT2 (and thence OPT1), our solution strategy is: First, determine the functional $\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{p}))$ by parametric analysis, and then minimize over $\boldsymbol{p} \in \mathbb{P}$ by exploiting the unique property of $\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{p}))$.

Note that the first step in effect involves a convex quadratic parametric programming (CQPP), which has been addressed in optimization and control community for sensitivity analysis and explicit controller design [183, 184]. Moreover, the study of solution path algorithms in our field [185, 186] can also be regarded as special cases of CQPP. Nonetheless, existing work on CQPP is technically insufficient, because (1) Due to the presence of the

[2]Then the induced matrix $\boldsymbol{Q} \triangleq \boldsymbol{K} \circ \boldsymbol{y}\boldsymbol{y}^T$ is also strictly positive, hence the optimization is strictly convex. For situations in which $\boldsymbol{K}$ is only positive semidefinite, a decomposition technique detailed in the supplementary material, can be used to reduce the problem to the strictly positive case.

constraint $\boldsymbol{\alpha}^T\boldsymbol{y} = 0$, the problem at hand corresponds to a "degenerate" case for which existing solution is still lacking. (2) Some important properties of the parametric solution, specifically its geometric structure, are not entirely revealed in prior works.

In the next section, we target the the inner minimization for parametric analysis. Our results not only provide the analytical form of the solution in critical regions (defined later), but also demonstrate that the overall learning problem (OPT2) is equivalent to a convex maximization.

4.3.3 The Equivalent Convex Maximization Problem

To begin with, the inner minimization is rewritten in a more compact form:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \mathcal{J}(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{Q}\boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\alpha} \\ \text{subject to} \quad & \boldsymbol{C}^\alpha\boldsymbol{\alpha} \leq \boldsymbol{C}^p\boldsymbol{p} + \boldsymbol{C}^0, \quad \boldsymbol{y}^T\boldsymbol{\alpha} = 0. \end{aligned} \tag{IO}$$

where $Q_{ij} = y_i\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)y_j$, and $\boldsymbol{C}^\alpha$, $\boldsymbol{C}^p$ and $\boldsymbol{C}^0$ are constant matrices encapsulating the constraints.

4.3.3.1 A Sufficient Condition for the Existence of Parametric Solution

We first demonstrate that, interestingly, a mild sample partition condition is sufficient for the existence and uniqueness of the parametric solution of (IO).

Definition 8. ***(Active Constraint)*** *After a solution of (IO) has been obtained as $\boldsymbol{\alpha}^*(\boldsymbol{p})$. The i^{th} row of the constraint is said to be active at $\boldsymbol{p}$, if $\boldsymbol{C}_i^\alpha\boldsymbol{\alpha}^*(\boldsymbol{p}) = \boldsymbol{C}_i^p\boldsymbol{p} + \boldsymbol{C}_i^0$, and inactive if $\boldsymbol{C}_i^\alpha\boldsymbol{\alpha}^*(\boldsymbol{p}) < \boldsymbol{C}_i^p\boldsymbol{p} + \boldsymbol{C}_i^0$. We denote the index set of active inequalities by $\mathcal{A}$, and inactive ones by $\mathcal{A}^C$. We use $\boldsymbol{C}_\mathcal{A}^\alpha$ to represent row selection of matrix $\boldsymbol{C}^\alpha$, i.e, $\boldsymbol{C}_\mathcal{A}^\alpha$ contains rows of $\boldsymbol{C}^\alpha$whose index is in $\mathcal{A}$.*

Definition 9. ***(Partition of Samples)*** *Based on the value of α_i at optimal, the i^{th} sample is called:*

- *Non-support vectors, denoted by $i \in \mathcal{O}$, if $\alpha_i^* = 0$.*
- *Unbounded support vectors, denoted by $i \in \mathcal{S}_u$ if we have strictly $0 < \alpha_i^* < c_i p_i$.*
- *Bounded support vectors, denoted by $\mathcal{S}_b$, if $\alpha_i^* = c_i p_i$.*

Definition 10. ***(Non-degeneracy by Sample Partition)*** *We say that a solution of (IO) is non-degenerate if the unbounded support vector set $\mathcal{S}_u$ contains at least one $\{i \mid y_i = +1\}$ and at least one $\{i' \mid y_i' = -1\}$*

Now we connect non-degeneracy, defined as a sample partition property of large margin learning, to the existence and uniqueness of the parametric solution.

Lemma 26. *If the solution $\boldsymbol{\alpha}^*$ of (IO) is non-degenerate, then*

- *The matrix $\boldsymbol{H} \triangleq \frac{\boldsymbol{Q}^{-1}\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{Q}^{-1}}{\boldsymbol{y}^T\boldsymbol{Q}^{-1}\boldsymbol{y}} - \boldsymbol{Q}^{-1}$ is negative semidefinite, and $\boldsymbol{R} \triangleq \boldsymbol{C}_{\mathcal{A}}^{\alpha}\boldsymbol{H}\boldsymbol{C}_{\mathcal{A}}^{\alpha T}$ is strictly negative definite, hence is invertible.*
- *The parametric solution $\boldsymbol{\alpha}^*(\boldsymbol{p})$ exists and is unique.*

Remark The invertibility of the matrix guarantees the uniqueness of the Lagrangian multipliers of (IO) and hence the existence and uniqueness of the parametric solution. The non-degeneracy condition is indeed a mild requirement: in fact in large margin learning formalism, the unbounded support vectors are essentially the sample points that lie on the decision boundaries, constructing the normal vector and the interception of the hyperplane. In practice to have meaningful classification this condition is a necessity and is expected to be satisfied.

4.3.3.2 Local Explicit Form of the Parametric Optimality

With the previous definitions and Lemma 4.5.2, the following theorem provides the explicit form of $\boldsymbol{\alpha}^*(\boldsymbol{p})$, as well as explicit "critical regions" in which the dependence stands.

Theorem 27. *Assume that the solution of (IO) is non-degenerate and induces a set of active and inactive constraints $\mathcal{A}$ and $\mathcal{A}^{\mathcal{C}}$, respectively. With $\boldsymbol{H}$, $\boldsymbol{R}$ defined previously and $\boldsymbol{T} \triangleq \boldsymbol{H}(\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T$, $\widetilde{\boldsymbol{e}} \triangleq \boldsymbol{C}_{\mathcal{A}}^{\alpha}\boldsymbol{H}\boldsymbol{1}$, we have*
(1) The optimal solution is a continuous piecewise affine function of $\boldsymbol{p}$. And in the critical region defined by

$$\begin{cases} \boldsymbol{R}^{-1}(\boldsymbol{C}_{\mathcal{A}}^{p}\boldsymbol{p} + \boldsymbol{C}_{\mathcal{A}}^{0} + \widetilde{\boldsymbol{e}}) \geq 0 \\ \boldsymbol{C}_{\mathcal{A}^C}^{p}\boldsymbol{p} + \boldsymbol{C}_{\mathcal{A}^C}^{0} - \boldsymbol{C}_{\mathcal{A}^C}^{\alpha}\boldsymbol{T}\boldsymbol{R}^{-1}(\boldsymbol{C}_{\mathcal{A}}^{p}\boldsymbol{p} + \boldsymbol{C}_{\mathcal{A}}^{0} + \widetilde{\boldsymbol{e}}) \geq 0 \end{cases} \tag{4.23}$$

the optimal solution $\boldsymbol{\alpha}^*$ of (IO) admits a closed form

$$\boldsymbol{\alpha}^*(\boldsymbol{p}) = \boldsymbol{T}\boldsymbol{R}^{-1}(\boldsymbol{C}_{\mathcal{A}}^{p}\boldsymbol{p} + \boldsymbol{C}_{\mathcal{A}}^{0} + \widetilde{\boldsymbol{e}}) \tag{4.24}$$

*(2) The optimal objective $\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{p}))$ is a **continuous** piece-wise quadratic (**PWQ**) function of $\boldsymbol{p}$.*

Remark The theorem indicates that each time the inner optimization (IO) is solved, full information in a well-defined neighborhood (critical region) can be retrieved as a function of the auxiliary variable. Hence one can efficiently calculate the closed form optimal solution and its gradient in that region, without having to solve (IO) again. (2) shows that $\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{p}))$ is continuous but non-smooth.

4.3.3.3 Global Structure of the Optimality

Recall that our goal is to solve $\max_{\boldsymbol{p}\in\mathbb{P}} \mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{p}))$. In this part, we show that the problem is equivalent to convex maximization by revealing several important geometric properties of $\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{p}))$ as a function of $\boldsymbol{p}$.

Theorem 28. *Still assuming non-degeneracy, then*
*(1) There is a finite number of critical regions $CR_1, \cdots, CR_{N_r}$ which constitute a **partition** of the feasible set of $\boldsymbol{p}$, i.e., each feasible $\boldsymbol{p}$ belongs to one and only one critical region.*
*(2) $\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{p}))$ is a globally **convex** function of $\boldsymbol{p}$, and is **almost everywhere differentiable**.*
*(3) $\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{p}))$ is **difference-definite**, i.e., the differences between its expressions on neighboring polyhedron critical regions have positive or negative semidefinite Hessian.*
(4) Let the common boundary of any two neighboring critical regions CR_i and CR_j be $\boldsymbol{a}^T\boldsymbol{p}+b$, then there exist a scalar β and a constant c, such that

$$\mathcal{J}_i(\boldsymbol{\alpha}^*(\boldsymbol{p})) = \mathcal{J}_j(\boldsymbol{\alpha}^*(\boldsymbol{p})) + \left[\boldsymbol{a}^T\boldsymbol{p} + b\right]\left[\beta\boldsymbol{a}^T\boldsymbol{p} + c\right].$$

Remark Although the number of critical regions is finite, in the worst case it could be exponential to the dimension of $\boldsymbol{p}$. Hence one cannot solve $\max_{\boldsymbol{p}\in\mathbb{P}} \mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{p}))$ by naively enumerating all possible critical regions. The globally convex PWQ property of $\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{\theta}))$ revealed by (2) is critical: now that the class of learning problem formulated in (OPT1) is equivalent to maximizing a non-smooth convex function, which is well known to be *NP-hard*. Fortunately, we will show in next section that there exists an optimality condition that can be exploited to design efficient global optimization algorithms. Lastly, (3) and (4) imply that the expressions of $\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{p}))$ on neighboring critical regions cannot be arbitrary, but is to some extent bounded. Those properties can be further harnessed for solution approximation.

4.3.4 Global Optimality Condition and Parametric Dual Maximization

To ease the notation, we hide the intermediate variable and denote

$$\mathcal{F}(\boldsymbol{p}) \triangleq \mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{p})) \tag{4.25}$$

then (OPT2) becomes $\max_{\boldsymbol{p}\in\mathbb{P}} \mathcal{F}(\boldsymbol{p})$. From the properties of $\mathcal{F}(\boldsymbol{p})$, or $\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{p}))$, given in Theorem 4.5.2 and Theorem 4.5.2, we know that the problem is in effect a *convex piece-wise quadratic maximization*. In this section, we propose a global optimization algorithm based on an optimality condition and a level set approximation technique.

4.3.4.1 A Global Optimality Condition

Several global optimality conditions for maximizing convex function, particularly convex quadratic functions, have been proposed before [187, 188]. In this work, we adapt a version

of Strekalovsky's condition for non-smooth case. First of all, the notion of level set is defined as the set of variables that produce the same function values, i.e.,

Definition 11. *The level set of the function $\mathcal{F}$ at $\boldsymbol{p}$ is defined by*

$$E_{\mathcal{F}(p)} = \{q \in \mathbb{R}^n \mid \mathcal{F}(q) = \mathcal{F}(p)\}$$

A sufficient and necessary condition for a point $\boldsymbol{p}^*$ to be the global maximizer of $\mathcal{F}(\boldsymbol{p})$ reads,

Theorem 29. *$\boldsymbol{p}^*$ is a global optimal solution of the problem $\max_{\boldsymbol{p}\in\mathbb{P}} \mathcal{F}(\boldsymbol{p})$, if and only if for all $\boldsymbol{p} \in \mathbb{P}$, $\boldsymbol{q} \in E_{\mathcal{F}(\boldsymbol{p}^*)}$, $g(\boldsymbol{q}) \in \partial\mathcal{F}(\boldsymbol{q})$, we have*

$$(\boldsymbol{p} - \boldsymbol{q})^T g(\boldsymbol{q}) \leq 0 \tag{4.26}$$

where $\partial\mathcal{F}(\boldsymbol{q})$ is the set of subgradients of $\mathcal{F}$ at $\boldsymbol{p}$.

By virtue of Theorem 4.5.3, we can verify the optimality of any point $\boldsymbol{p}$ by solving

$$\Delta(\boldsymbol{p}) \triangleq \max_{\substack{\boldsymbol{q}\in E_{\mathcal{F}(\boldsymbol{p})},\ \boldsymbol{p}'\in\mathbb{P} \\ g(\boldsymbol{q})\in\partial\mathcal{F}(\boldsymbol{q})}} (\boldsymbol{p}' - \boldsymbol{q})^T g(\boldsymbol{q}) \tag{4.27}$$

and checking if $\Delta(\boldsymbol{p}) \leq 0$. We call the above maximization the *auxiliary problem* at $\boldsymbol{p}$. The major difficulty is that the level set $E_{\mathcal{F}(\boldsymbol{p})}$ is hard to calculate explicitly. Next, we study solution method for (4.27) by approximating the level set with a collection of representative points.

4.3.4.2 Approximate Level Set

Definition 12. *Given a user specified approximation degree m, the approximation level set for $E_{\mathcal{F}(\boldsymbol{p})}$ is defined by*

$$A_{\boldsymbol{p}}^m = \left\{\boldsymbol{q}^1, \boldsymbol{q}^2, \cdots, \boldsymbol{q}^m \mid \boldsymbol{q}^i \in E_{\mathcal{F}(\boldsymbol{p})}\ i = 1, 2, \cdots, m\right\}$$

Consider solving the auxiliary problem approximately by replacing $E_{\mathcal{F}(\boldsymbol{p})}$ with $A_{\boldsymbol{p}}^m$, then for each $\boldsymbol{q}^i$, (4.27) becomes

$$\max_{\boldsymbol{p}\in\mathbb{P},\ g(\boldsymbol{q}^i)\in\partial\mathcal{F}(\boldsymbol{q}^i)} (\boldsymbol{p} - \boldsymbol{q}^i)^T g(\boldsymbol{q}^i) \tag{4.28}$$

Since $\mathcal{F}(\boldsymbol{p})$ is almost everywhere differentiable, in most cases $g(\boldsymbol{q}^i)$ is unique and equals to the gradient $\nabla\mathcal{F}(\boldsymbol{q}^i)$. Then the auxiliary problem is a simple *linear program*. In the cases when $\boldsymbol{q}^i$ is on the boundary of critical regions, $\partial\mathcal{F}(\boldsymbol{q}^i)$ becomes a convex set, and the auxiliary problem becomes a bilinear program. General bilinear programs are hard, but fortunately (4.78) has disjoint feasible sets, and one can show that

Algorithm 4: Parametric Dual Maximization

Choose $\boldsymbol{p}^{(0)} \in \mathbb{P}$; set $k = 0$; compute $\boldsymbol{p}_*$ with subgradient descent.
while $k \leq iter\_max$ **do**
  Starting from $\boldsymbol{p}^{(k)}$, find a local maximizer $\boldsymbol{r}^{(k)} \in \mathbb{P}$ with a local solver.
  Construct $A^m_{\boldsymbol{r}^{(k)}}$ at $\boldsymbol{r}^{(k)}$ by (4.31) (4.95)
  Solve (IO) if a new critical region is encountered, otherwise use (4.47).
  for $\boldsymbol{q}^i \in A^m_{\boldsymbol{r}^{(k)}}$ **do**
    for $\boldsymbol{g}^j \in V(\partial\mathcal{F}(\boldsymbol{q}^i))$ **do**
      Solve linear programming $\boldsymbol{u}_{ij} = \text{argmax}_{\boldsymbol{p}\in\mathbb{P}}\ (\boldsymbol{p} - \boldsymbol{q}^i)^T\boldsymbol{g}^j$
    Let $j^* = \text{argmax}_j\{\boldsymbol{u}_{ij}\}$; $(\boldsymbol{u}^i, \boldsymbol{s}^i) = (\boldsymbol{u}_{ij^*}, \boldsymbol{g}^{j^*})$
  Let $i^* = \text{argmax}_i\{(\boldsymbol{u}^i - \boldsymbol{q}^i)^T\boldsymbol{s}^i\}$; $\boldsymbol{u}^{(k)} = \boldsymbol{u}^{i^*}$
  if $(\boldsymbol{u}^{i^*} - \boldsymbol{q}^{i^*})^T\boldsymbol{s}^{i^*} > 0$ **then**
    Set $\boldsymbol{p}^{(k+1)} = \boldsymbol{u}^{(k)}$; $k = k + 1$; # improvement found.
  else
    Terminate and output $\boldsymbol{p}^{(k)}$; # optimality checked
  Collecting explored critical region and explicit forms given in (4.47)(4.46).

Proposition 30. *Problem (4.78) is equivalent to*

$$\max_{\boldsymbol{p}\in\mathbb{P}} \left\{ \max_{g(\boldsymbol{q}^i)\in V(\partial\mathcal{F}(\boldsymbol{q}^i))} (\boldsymbol{p} - \boldsymbol{q}^i)^T g(\boldsymbol{q}^i) \right\} \tag{4.29}$$

which indicates that the optimal solution to (4.78) must be on the vertex of the feasible polyhedron. As such, (4.78) can be expanded into a set of linear programs, each of which is substantiated by an element in $A^m_{\boldsymbol{p}}$ and a vertex of $\partial\mathcal{F}(\boldsymbol{q}^i)$.

4.3.4.3 The PDM Algorithm

With the approximate auxiliary problem solved, we can immediately determine if an *improvement* can be made at the current $\boldsymbol{p}$. More specifically, let $\{(\boldsymbol{u}^i, \boldsymbol{s}^i),\ i = 1, \cdots, m\}$ be the solution of (4.78) on $A^m_{\boldsymbol{p}}$, i.e.,

$$(\boldsymbol{u}^i - \boldsymbol{q}^i)^T\boldsymbol{s}^i = \max_{\boldsymbol{p}\in\mathbb{P},\ g(\boldsymbol{q}^i)\in V(\partial\mathcal{F}(\boldsymbol{q}^i))} (\boldsymbol{p} - \boldsymbol{q}^i)^T g(\boldsymbol{q}^i) \tag{4.30}$$

and define $\Delta(A^m_{\boldsymbol{p}}) = \max_{i=1,\cdots,m}(\boldsymbol{u}^i - \boldsymbol{q}^i)^T\boldsymbol{s}^i$. Then with the convexity of $\mathcal{F}$ we have

Proposition 31. *For any $\boldsymbol{p} \in \mathbb{P}$, if there exist $\boldsymbol{q}^i \in A^m_{\boldsymbol{p}}$, $g(\boldsymbol{q}^i) \in V(\partial\mathcal{F}(\boldsymbol{q}^i))$, and $\boldsymbol{u}^i$ defined in (4.80), such that $(\boldsymbol{u}^i - \boldsymbol{q}^i)^T g(\boldsymbol{q}^i) > 0$, then we must have $\mathcal{F}(\boldsymbol{u}^i) > \mathcal{F}(\boldsymbol{p})$.*

Now the remaining work is to construct the approximate level set given the current $\boldsymbol{p}$ and the degree m. The following lemma shows that this is possible if a global minimizer is available.

Lemma 32. *Let the global* ***minimizer*** *of* $\mathcal{F}(\boldsymbol{p})$ *be* $\boldsymbol{p}_*$*, then for* $\boldsymbol{p} \neq \boldsymbol{p}_*$ *and* $\boldsymbol{h} \in \mathbb{R}^n$*, there exist a* ***unique*** *positive scalar* γ*, such that* $\boldsymbol{p}_* + \gamma \boldsymbol{h} \in E_{\mathcal{F}(\boldsymbol{p})}$.

With this guarantee, we write the approximate level set as

$$A_{\boldsymbol{p}}^m = \left\{ \boldsymbol{q}^1, \boldsymbol{q}^2, \cdots, \boldsymbol{q}^m \mid \boldsymbol{q}^i = \boldsymbol{p}_* + \gamma_i \boldsymbol{h}^i \in E_{\mathcal{F}(\boldsymbol{p})} \right\} \tag{4.31}$$

To explore directions for improvement, a natural choice of $\boldsymbol{h}$ is a set of orthogonal bases. Specifically, we could start with a random $\boldsymbol{h}^1$ and use the Gram-Schmidt algorithm to extend it to m orthogonal basis. For each $\boldsymbol{h}^i$, the corresponding γ_i is found by solving:

$$\Phi(\gamma_i) \triangleq \mathcal{F}(\boldsymbol{p}_* + \gamma_i \boldsymbol{h}^i) - \mathcal{F}(\boldsymbol{p}) = 0 \tag{4.32}$$

As stated in Lemma 4.5.3, the above function has a unique root, which can be computed efficiently with line searching method. To obtain the global minimizer, we have to solve $\boldsymbol{p}_* = \text{argmin}\mathcal{F}(\boldsymbol{p})$, which is a convex minimization problem. Using Theorem 4.5.2, we show (in supplementary material) that a sub-gradient descent method with T iterations converges to the global minimum within $O(1/\sqrt{T})$.

Organizing all building blocks developed so far, we summarize the PDM procedure in Algorithm 1. Given the current solution $\boldsymbol{p}^{(k)}$, the algorithm first tries to improve it with existing methods such as AO, CCCP, SGD, etc. After finding a local solution $\boldsymbol{r}^{(k)}$, the approximate level set $A_{\boldsymbol{r}^{(k)}}^m$ is obtained by solving (4.95) and constructing (4.31). With $A_{\boldsymbol{r}^{(k)}}^m$ and the current sub-gradient, one or several linear program is solved to pick up the vector $\boldsymbol{u}^{(k)}$ that maximizes the condition (4.69) of Theorem 4.5.3. If this maximal value, i.e., $\Delta(A_{\boldsymbol{p}}^m)$, is greater than 0, then by Proposition 31, $\boldsymbol{u}^{(k)}$ must be a strictly improved solution compared to $\boldsymbol{r}^{(k)}$. As such, the algorithm continues with $\boldsymbol{p}^{(k+1)} = \boldsymbol{u}^{(k)}$. Otherwise if $\Delta(A_{\boldsymbol{p}}^m) \leq 0$, the algorithm terminates since no improvement could be found at the current point with the user specified approximation degree. For convergence, we have

Theorem 33. *Algorithm I generates a sequence* $\{\boldsymbol{p}^{(1)}, \cdots, \boldsymbol{p}^{(k)}, \cdots\}$ *having non-decreasing function values. The sequence converges to an approximate maximizer of* $\mathcal{F}(\boldsymbol{p})$ *in a finite number of steps.*

In each iteration, we only have to solve $m|V(\partial\mathcal{F}(\boldsymbol{q}^i))|$ linear programs, and in most cases $|V(\partial\mathcal{F}(\boldsymbol{q}^i))| = 1$ due to the almost everywhere differentiability shown in Theorem 4.5.2. When constructing the approximiate level set, we need to solve at most m convex quadratic programs (IO)s, which seems computationally expensive. However, note that this problem resembles the classic SVM dual, where a variety of existing methods can be reused for acceleration [189]. Moreover, by virtue of the optimality structure revealed in Theorem 4.5.2 and 4.5.2, a list of explored critical regions and the corresponding explicit optimalities can be stored. If the current $\boldsymbol{p}$ is on this list, all information could be retrieved in an explicit form, and there is no need to solve the quadratic problem again. To further accelerate the algorithm, one can "enlarge" critical regions. See supplementary material for a discussion.

4.4 Experiment

In this section, we first report optimization and generalization performance of PDM for the training of S^{3}VM and VCMKL, and then elaborate two case studies, one involves using CPLM for optimal control and the other using HS^{3}M for event detection. The source code and data sets can be found at https://github.com/Yuxun/PDM.

4.4.1 Optimization and Generalization Performance

The purposes of this experiment are three folds: (1) Test the proposed PDM as a novel optimization algorithm for machine learning and compare it to other state-of-the-art optimization techniques. In particular we focus on the training of the popular semi-supervised learning paradigm S^{3}VM and the proposed VCMKL. (2) Justify the effort of improving local solutions (approaching global optimum), by comparing the generalization performance in terms of testing accuracy. (3) Test CPLM and VCMKL as enhanced classification methods for imbalanced data or data with hidden subgroups. To begin with, we introduce some experimental setup.

4.4.1.1 Datasets and Experiment Setup

Details of the data sets used in this experiment are listed in Table 4.4. For S^{3}VM, we report results on four popular data sets for semi-supervised learning, i.e., *2moons* (D1), *coil* (D2), *robot* (D3) and *2spiral* (D4, with simulator). In each experiment, 60% of the samples are used for training, in which only a small portion are assumed to be labeled samples. 10% of the data are used as a validation set for choosing hyperparameters. With the remaining 30%, we evaluate the generalization performance. For VCMKL we adopt the same training, validation and testing partition on *Vowel* (D5), *Music* (D6), *Bank* (D7) and *Wave* (D8, with simulator) data sets. To create a latent data structure, we assume that only grouped binary labels are known.

The Gaussian kernel $\kappa(\boldsymbol{x}, \boldsymbol{y}) = \exp\{\|\boldsymbol{x} - \boldsymbol{y}\|^2/2\sigma^2\}$ is used for all experiments. Following model selection suggestions [182][168], best hyperparamter combination C_1, C_2, σ^2 are chosen with cross validation from $C_1 \in \{10^{0:0.5:3}\}$, $\sigma^2 \in \{(1/2)^{-3:1:3}\}$ and $C_2 \in \{10^{-8:1:0}\}$ for S^{3}VM and $C_2 \in \{10^{-4:1:4}\}$ for VCMKL. A simple gradient ascent is used as the local minimizer for PDM. All experiments are conducted on a workstation with Dual Xeon x5687 CPUs and 72GB memory.

Table 4.1: Data sets. D4-D3 for S^{3}VM and D5-D8 for VCMKL

Data set	ID	# classes	# samples	# features	labeled
2moons	D1	2	200	2	2
Coil	D2	3	216	1024	6
Robot	D3	4	2456	25	40
2spiral	D4	2	100000	2	4
Vowel	D5	10	990	11	grouped
Music	D6	10	2059	68	grouped
Bank	D7	9	7166	649	grouped
Wave	D8	30	100000	40	grouped

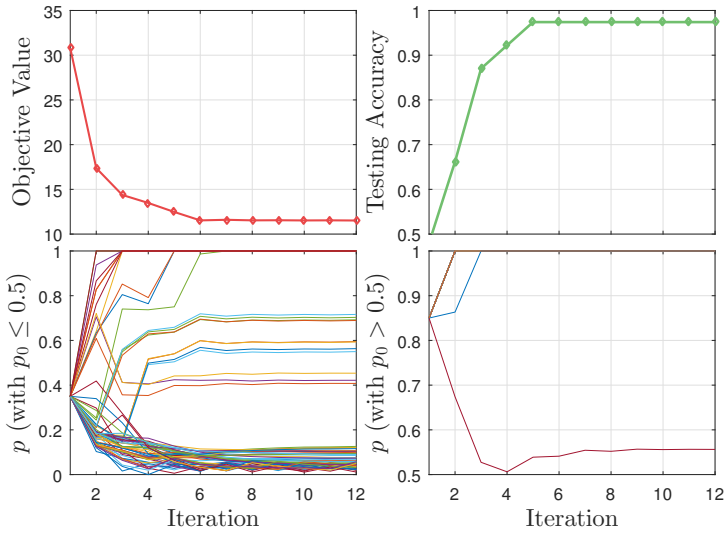


Figure 4.4: PDM in each iteration for S^{3}VM training. Randomized initiation; $m = 20$; D1 dataset

4.4.1.2 Demo: Iterative Results of PDM

To get more intuition on how PDM works, we use PDM to train S^{3}VM on the D1 dataset, and plot the iterative evolution of objective function, testing accuracy and the values of $\boldsymbol{p}$ in Figure 4.4. The approximation level m is set to $0.1length(p) = 20$, and the initial $p^{(0)}$ is chosen randomly. We observe that PDM converges within 12 iterations (top left subfigure). The testing accuracy increases from 48% to above 98% (top right subfigure), showing improvements in both optimization and generalization performance. Moreover, the

Table 4.2: Normalized objective value (OPT1. First row for each dataset. The lower the better). Time usage (Second row for each dataset. $s = seconds; h = hours$)

	Data	GD	CCCP	AO	LCS	IA	BB	PDM1	PDM2
S^3VM	D1	2.39	2.82	4.83	5.55	1.79	**1.00**	1.03	**1.00**
		1.7s	6.2s	2.7s	6.7s	3.4s	210s	16s	35s
	D2	3.74	3.92	3.46	4.98	2.35	**1.00**	1.19	1.03
		5.3s	6.8s	**4.3s**	7.9s	5.6s	362s	43s	83s
	D3	3.95	4.23	3.48	6.96	2.85	*	1.11	**1.00**
		33s	56s	28s	43s	**27s**	*	231s	489s
	D4	6.98	4.91	4.90	6.16	4.22	*	1.31	**1.00**
		0.19h	0.41h	0.33h	0.37h	0.46h	*	1.4h	2.7h
VCMKL	D5	4.45	5.31	4.85	4.09	*	*	1.13	**1.00**
		26s	54s	33s	68s	*	*	209s	451s
	D6	6.51	5.34	4.77	6.82	*	*	1.28	**1.00**
		63s	90s	72s	101s	*	*	468s	997s
	D7	6.78	7.69	4.17	6.22	*	*	1.26	**1.00**
		326s	371s	**263s**	477s	*	*	1217s	2501s
	D8	10.2	5.16	6.35	7.57	*	*	1.54	**1.00**
		0.23h	0.73h	0.66h	0.93h	*	*	2.5h	4.8h

auxiliary variable $\boldsymbol{p}$ approaches global optimum even with random initial values (bottom subfigures). Note that in this process, a total number of 36 (IO)s are solved and about 2/3 of the critical regions have been reused more than once.

4.4.1.3 Optimization and Generalization Performance

We next compare PDM with different optimization methods in terms of their optimization and generalization performance. The algorithms considered for S^3VM training are: Gradient Descent (GD) in [176], CCCP in [190], Alternating Optimization (AO) in [191], Local Combinatorial Search (LCS) in [167], Infinitesimal Annealing (IA) in [192], Branch and Bound (BB) in [177]. The algorithms included for VCMKL are GD in [146], CCCP in [169], AO in [143], adapted LCS in [167]. The proposed PDM is tested with two versions by setting the approximation degree $m = 0.1length(p)$ (PDM1) and $m = 0.2length(p)$ (PDM2).

In Table 4.6, objective function values of OPT1 (normalized by the smallest one) are shown in the upper row, and the corresponding computation times are given in the second

Table 4.3: Generalization Performance (error rates). Averaged over 10 random data partitions. Error rate greater than or close to 50% should be interpreted as "failed".

	Data	GD	CCCP	AO	LCS	IA	BB	PDM1	PDM2
S^3VM	D1	51.4	60.0	52.8	65.5	37.5	**0.0**	1.9	0.2
	D2	57.9	66.1	47.9	61.1	57.2	**0.0**	5.3	1.1
	D3	26.6	29.3	59.8	38.8	27.4	*	9.5	**3.3**
	D4	52.1	39.8	40.0	45.4	31.4	*	3.5	**2.0**
VCMKL	D5	15.8	16.2	13.5	9.9	*	*	2.5	**1.7**
	D6	39.8	43.7	40.8	39.4	*	*	12.1	**7.6**
	D7	20.0	19.4	19.8	22.5	*	*	8.9	**5.1**
	D8	53.1	36.7	39.7	46.2	*	*	19.9	**13.1**

row for each data. Note that although BB provides exact global optimum for small data set D1 and D2, it runs out of memory (72GB!) for other datasets due to the exponential growth of its search tree. On the other hand, PDM1&2 provides a near optimal solution to BB with much less time and space usage. For larger data sets (D4-D8) on which BB can not be executed, PDM outperforms all the other local optimization methods: We observe that PDM achieves a significantly improved objective value, and the runner up is at least 2.8 times larger. Although the running time is longer than local methods, PDM is still scalable (D4 & D8 have 10^5 samples), hence can be carried out for large scale problems.

In Table 4.3, we compare the generalization performance of different algorithms in terms of testing error rate. It appears clearly that the global optimal solution provided by BB and PDM has excellent generalization error rate, while other local optimization methods perform much worse, and even fail completely (e.g., on D1, D2, D4, D8). This observation is consistent with previous findings [177] [182], justifying the extra computational overhead required to pursue the global optimum.

4.4.1.4 The Effect of Approximation Degree and Number of Kernels

Comparing PDM1 and PDM2 in Table 4.6&4.3, we note that in general, increasing the approximation degree m will produce better optimization and generalization performance. To investigate the effect of m, we use PDM to train S^3VM on D3, and plot in Figure 4.5 the optimum value, testing accuracy, time and space usage as a function of m (from 80 to 650). It appears that further increasing m after some large enough value (e.g., 300 in Figure 4.5) only provide marginal improvement in both training and testing. Also, seeing that the computational time usage grows (slightly) super-linearly and that the space usage grows almost linearly, we suggest using an $m \in [0.1length(p), 0.2length(p)]$, a tradeoff between

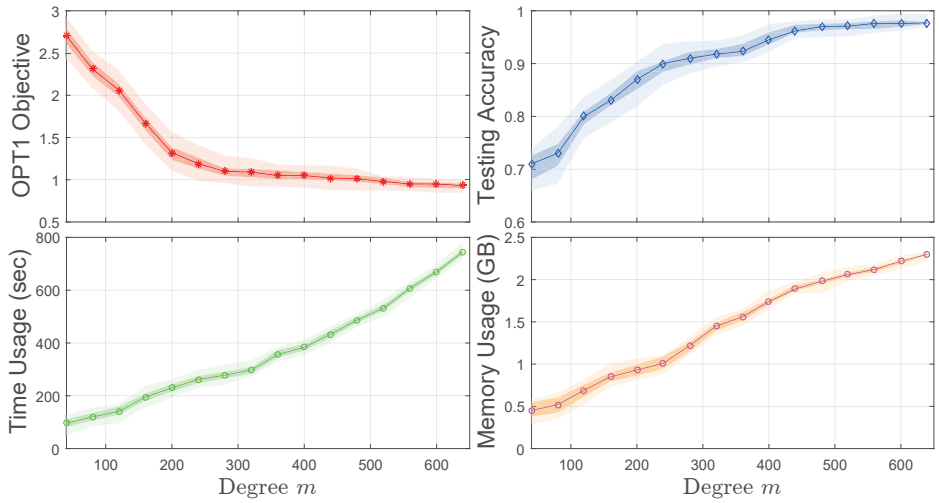


Figure 4.5: The effect of m for PDM. D3 Dataset; Average and CIs for 50 runs.

training/testing accuracy and computational overhead.

Two cases are used to demonstrate the effect of number of kernels M on the classification performance of VCMKL. Note again that labels of the raw data are transformed into a binary case, e.g. for the vowel data, $y = +1$ if the label is 'hOd' or 'hod', and $y = -1$ for the rest of 8 classes. Figure 4.6 shows the testing accuracy versus the number of kernels with 3 commonly used kernel families (linear, polynomial with different orders and RBF with various σ). Interestingly, in all cases it is seen that the testing accuracy is improved in the first few steps as M increases, however, further combining more kernels does not help, or even lead to a degraded performance due to overfitting (for linear and nonlinear kernels, respectively). In particular, the saturated point for linear kernels is just the number of hidden subgroups, i.e. $M = 4$ for D5 and $M = 8$ for D6 data, while for nonlinear kernels the optimal M is smaller as in the transformed space the subgroups may merge. This observation not only justifies the veto-consensus intuition to describe hidden subgroups, but also is consistent with the theoretical analysis, which provides additive upper bound and does not encourage the use of many kernels.

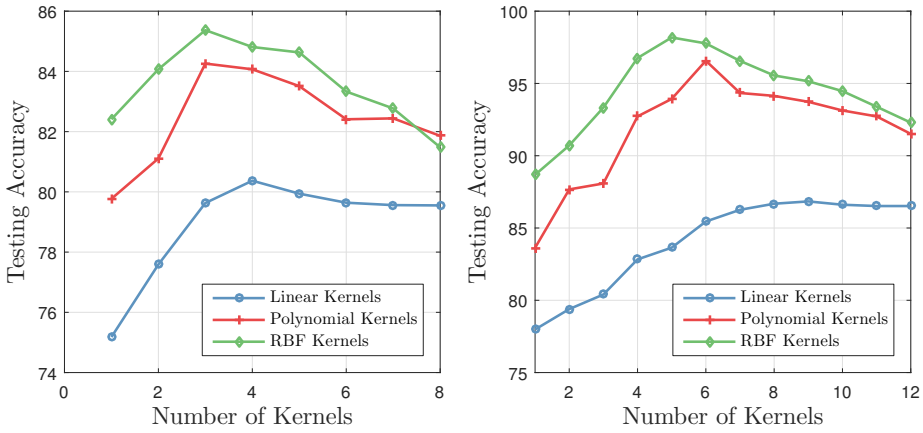


Figure 4.6: Testing Accuracy vs. M number of kernels. Left:D5, Right:D6

4.4.2 Case Study: CPLM Based User Comfort Learning for HVAC Model Predictive Control (MPC)

We show a concrete use case of the proposed "learning for application" scheme for optimal control purpose. In particular, an integration of CPLM based thermal comfort learning and HVAC MPC is demonstrated in this section: We first provide a brief introduction about comfort-aware HVAC MPC, and then verify the CPLM learning method based on real-world thermal comfort data set. Finally we conduct an MPC simulation to show that comfort-aware and cost-effective HVAC operation can be achieved by the integration of the dedicated machine learning method with optimization based control strategy.

4.4.2.1 Integration of CPLM Based Comfort Zone Learning and HVAC MPC

Buildings accounts for nearly 40% of global energy consumption, with variations among countries about half of this energy is devoted to indoor climate regulations via Heating, Ventilation and Air Conditioning (HVAC) systems [11]. Since the indoor environment has tremendous impact on occupants' health and productivity and should not be compromised, it is of fundamental importance to optimally control HVAC systems to decrease energy consumption while at the same time maintaining occupants' comfort preference. In this context, two lines of research, namely advanced control technologies for HVAC operation and modeling techniques for comfort requirement, have been motivated. With regard to HVAC control, a great deal of progress has been made from the control and automation community to reform the classic rule based control strategies. Recently model based optimal control schemes, especially the adaptations of Model Predictive Control (MPC) has achieved significant improvement in terms of both energy efficiency and demand response.

The basic idea of MPC (receding horizon) based synthesis is to optimize over a finite time horizon to take future effect into account, while only implementing the decisions of the current time slot. A typical optimization problem for MPC at each step can be written

as minimizing a cost function subjects to constraints such as system dynamics, initial/final state requirement, state operation requirement, etc. i.e.

$$\min J = \Phi(x_0, T_0, x_f, T_f) + \sum_{t=T_0}^{T_f} \phi(x_t, u_t, t) \tag{4.33}$$

$$\text{s.t.} \quad x_{t+1} = f(x_t, u_t, t) \tag{4.34}$$

$$\varphi(x_0, T_0, x_f, T_f) = 0 \tag{4.35}$$

$$\rho(x_t, u_t, t) \leq 0 \tag{4.36}$$

In the case of HVAC MPC, the objective function 4.33 is usually the total energy consumption of the HVAC system in the near future (24 hours for instance), the system equation 4.34 is the building thermal dynamics, and the operation constraint 4.36 is the human comfort requirement as a function of environmental and individual variables. Detailed development of system models for HVAC MPC is beyond the scope of this work, but can be found in literature [193]. For experimental purpose an adaptation of the existing work [141] is used for system dynamics. More specifically, the thermal dynamics of a room is described by the following heat balance equations:

$$\begin{aligned} m_{ai} T_{ai}^{t+1} = {} & N_i^t Q_p + h_w A_i (T_{out}^t - T_{ai}^t) \\ & + G_{fau,i}^t T_{fau}^t + G_{fcu,i}^t T_{fcu,i}^t + G_{nv,t}^t T_{out}^t \\ & + T_{ai}^t \left[m_{ai} - (G_{fau,i}^t + G_{fcu,i}^t + G_{nv,t}^t) \right] \end{aligned}$$

where $m_{ai} = \rho V$ is the air mass in room i, $Q_p = 3.48 * 10^5 J/h$ is the heat emission per person. $h_w = 6.12 * 10^3 J/m^2/h$ Similarly, the relative humidity also evolves according to the following conservation law:

$$\begin{aligned} m_{ai} H_{ai}^{t+1} = {} & N_i^t H_p \\ & + G_{fau,i}^t H_{fau}^t + G_{fcu,i}^t H_{fcu,i}^t + G_{nv,t}^t H_{out}^t \\ & + H_{ai}^t \left[m_{ai} - (G_{fau,i}^t + G_{fcu,i}^t + G_{nv,t}^t) \right] \end{aligned}$$

To calculate the energy consumption, consider the Enthalpy difference between inlet air and outlet air:

$$\begin{aligned} E_{fau,i}^t = {} & G_{fau,i}^t \left[C_a T_{out}^t + H_{out}^t (2500 + 1.84 T_{out}^t) \right] \\ & - G_{fau,i}^t \left[C_a T_{fau}^t + H_{fau}^t (2500 + 1.84 T_{out}^t) \right] \end{aligned}$$

where $C_a = 1.297 * 10^3 J/m^3/K$. As for fan power consumption, the following relation is usually used:

$$P_{fau}^t = P_{fau,rated} \left[\sum_{i=1}^{I} G_{fau,i}^t / G_{fau,rated}^t \right]^3$$

When it comes to comfort modeling, perhaps the most widespread one is the Predicted Mean Vote (PMV) [194]. The PMV model calculates an average thermal sensation index with four environmental variables and two personal variables by iteratively solving a series of heat balance equations . In the view that PMV is computationally expensive and not adaptive for other comfort related factors, researchers have been investigating data driving approaches from a machine learning perspective. Quite a few literature suggests the application of Artificial Neural Network for comfort learning and estimation [195] [196], while other existing supervised learning methods also have been explored, such as Support Vector Machine (SVM) with radius basis kernel [197] [198] and locally weighted regression models [199].

Although each of the two research disciplines has made remarkable contributions to the study of indoor environment regulation, there is a non-negligible gap between them. In fact in all of the aforementioned HVAC optimal control literature only box constraints on the environmental variables are considered, which independently specify the range of air temperature, relative humidity, etc. On the other hand the comfort models proposed in the previous research are only applied to simple feedback control schemes. The reason for this gap is understandable from a technical perspective: Since the MPC for HVAC control already requires solving a challenging large scale optimization problem, the incorporation of classic learning based comfort models will induce non-linearly coupled constraints for the environmental variables, making the corresponding optimization very hard (if not intractable) to solve numerically.

The CPLM proposed in this work bridges the gap between learning and control. The learned individual comfort models are essentially a set of linear inequalities. The advantage is immediate for the control side: The set of linear inequalities can be directly plugged into any optimization without increasing the inherent complexity of the problem. In the view of this, the solution we proposed here is a realization of "learning for application", in which a learning machine is justified not only by its classification performance, but also its compatibility with downstream applications.

The integration of the proposed comfort learning and HVAC optimal control in sensor rich smart buildings is shown in Figure 4.7. With the development of sensor network and information technology, the observability of building environmental states is greatly enhanced [200] [201] and user preference data could be easily collected with online survey tools [202] [203]. The calibrated HVAC system model, building thermal dynamics, and the learned comfort zone constitute the input of the MPC algorithm, which optimizes over future horizons and decide current settings for decision variables. The low level controller takes orders from the MPC algorithm and realizes the variable configuration with simple control laws such as PID. The framework in Figure 4.7 can be readily extended to many other control problems involving data-driven constraints, such as sensor network aided manufacturing process control, unmanned vehicle control, behavior related optimal economic mechanism design.

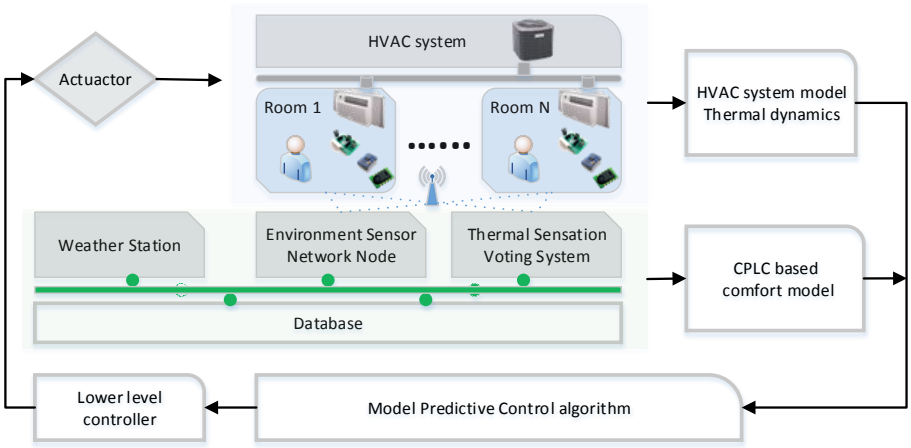


Figure 4.7: Integration of CPLM and HVAC optimal control in sensor rich smart buildings

4.4.2.2 Comfort Zone Learning with CPLM

Firstly, we test the classification performance of CPLM and compare it to other alternatives. The publicly available RP-884 data set [204] is used to train CPLM based comfort model. The data set is established as a part of ASHREA project for developing model of thermal comfort preference, and is collected through a series of field studies that covers 160 locations worldwide. Each row of the data set contains measurements of environmental variables (air temperature, radiant temperature, relative humidity, air velocity), individual physiological conditions (clothing, metabolic rate), as well as comfort sensation records. We use this data to reflect the expected thermal sensation of "average" person for control purposes [195] [198]. To eliminate the heterogeneity among different climate zones, a subset of the data originating from similar areas is selected, which yields 2839 labeled samples.

We compare the proposed CPLM with other popular machine learning methods, including a cost sensitive version of 2ν-SVM with Gaussian RBF and linear kernel, one class SVM, Deep Neural Network, AdaBoost and Lasso Logistic Regression. In order to justify the benefit of using Bayes consistent hinge loss, we also add the CPLM with naive cost sensitive loss into the comparison. Again 10 fold cross validation is performed for the selection of hyper-parameters in each algorithm. Table 4.4 shows the overall testing costs and the rankings of different methods with the ratio $r = c_2 : c_1$ ranging from 1 to 9. We see that when the cost ratio is large ($r \geq 5$), CPLM outperforms all the other methods. In particular, for $r \geq 7$ the performance improvement is more than **9.2%** compared to the runner-up. Together with a much better cost sensitive performance than the naive CPLM, the proposed large margin formulation and the use of cost sensitive hinge loss are supported. When the cost ratio is small, albeit an inferior performance than 2ν-RBF-SVM or deep neural network, CPLM is still better than the other methods. It should be restated that although some non-linear methods may perform better in terms of classification errors, they are depreciated for optimal

Table 4.4: Testing Cost Comparison

Method	**Testing Cost** with r = false positive cost : false negative cost = $c_2 : c_1$						
	1 : 1	2 : 1	3 : 1	4 : 1	5 : 1	6 : 1	7 : 1
CPLC	0.146 (3)	0.212 (3)	0.256 (2)	0.282 (2)	0.318 (1)	0.374 (1)	0.412 (1)
naive-CPLC	0.146 (4)	0.216 (4)	0.266 (4)	0.318 (3)	0.372 (4)	0.426 (3)	0.474 (4)
2ν-RBF-SVM	0.126 (2)	0.180 (1)	0.234 (1)	0.288 (3)	0.342 (3)	0.396 (2)	0.450 (2)
Linear SVM	0.205 (7)	0.366 (5)	0.527 (7)	0.560 (6)	0.598 (8)	0.637 (8)	0.675 (8)
One Class SVM	0.444 (8)	0.510 (8)	0.524 (6)	0.528 (5)	0.532 (5)	0.536 (5)	0.540 (5)
Deep NN	0.120 (1)	0.185 (2)	0.258 (3)	0.278 (1)	0.331 (2)	0.444 (4)	0.470 (3)
Ada Boost	0.202 (5)	0.370 (6)	0.460 (5)	0.576 (8)	0.582 (7)	0.608 (7)	0.634 (7)
Lasso LR	0.202 (6)	0.374 (7)	0.544 (8)	0.564 (7)	0.574 (6)	0.580 (6)	0.582 (6)

control purposes. In fact, among these methods only linear SVM and CPLM can be directly used for optimization-based applications without causing any extra difficulties, while CPLM outperforms linear SVM by **at least 25%** for all cost ratios.

4.4.2.3 The Impact of Learned Comfort Zone on HVAC MPC

To test the integrated framework we consider a typical room in an office building. The room is 20.1 meters long, 10.2 meters wide, and 4 meters high. It has four windows facing south and its HVAC system is equipped with one Fan Coil Unit (FCU) and one Fresh Air Unit (FAU). For the MPC we take 1 hour as time resolution and the next 24 hours as receding horizon. Air temperature and relative humidity setting points are chosen as decision variables.

In order to compare the effect of CPLM based comfort models with the box constraints used in traditional HVAC MPC work, other environmental and individual variables are presumed with average values. Hence the comfort requirement of each occupant reduces to a region in the 2D temperature-relative humidity space. We assume the room is occupied from 7am to 9pm with a maximum of 26 occupants and the number of occupants in each hour is generated according to users' working schedules. The cost ratio of false negative (classify "not comfort" as "comfort") and false positive (classify "comfort" as "not comfort") is taken to be $7 : 1$ [205]. Since HVAC systems can operate in different modes, two cases with *cold/dry* and *hot/humid* outdoor weather conditions are studied.

In each case we compare:

- **MPC1** : HVAC MPC operation with box constraints as comfort requirement.
- **MPC2** : HVAC MPC operation with online learning of CPLM as comfort requirement.

Condition I: Heating Modes with Cold and Dry Weather

In this setting, 48 hours (0am 21/Oct to 0am 23/Oct 2016) weather prediction data (historical record) for Beijing is used as "future" values of outdoor temperature and humidity. The HVAC MPC is operated for the first 24 hours and we assume that the variable setting generated by MPC could be realized at a much faster rate than the time resolution (1 hour).

In the occupied period (7am to 9pm) the occupants' comfort models are included as additional constraints for HVAC MPC. Figure 4.8 shows the learned CPLM of one user at the end of the control horizon and the box comfort zone. We observe that the box zone is too conservative for higher temperature and ignores possible comfort points (temperature and humidity combinations lying in the region C), while in region A, the ASHREA comfort requirement is violated as human usually feels colder when the humidity is low. On the other hand CPLM builds piece-wise linear boundaries and allows a much better description of thermal comfort.

As is discussed before, the learned CPLM is just a series of linear constraints and could be directly used to replace box constraints on temperature and humidity ($22 \leq T \leq 26$, $0.3 \leq H \leq 0.7$) by

$$\boldsymbol{w}_i^T \begin{bmatrix} T \\ H \\ 1 \end{bmatrix} \geq 0 \quad \forall i \in \{1, \cdots, M\}$$

without causing any extra difficulties for the optimization in the MPC. The HVAC operation results for two kinds of comfort specifications are compared in Figure 4.9 and Figure 4.10. We observe that during the period 11am to 5pm, MPC2 resulted in a higher (about 1 degree) room temperature while maintaining almost the same room humidity. As a consequence it consumes more energy for heating. Although MPC1 is slightly (about 4%) more efficient in terms of energy usage, the operated room temperature and humidity lie in the bottom left corner (region A in Figure 4.8) of the box comfort zone, which is too cold and dry and is **inadmissible** for users.

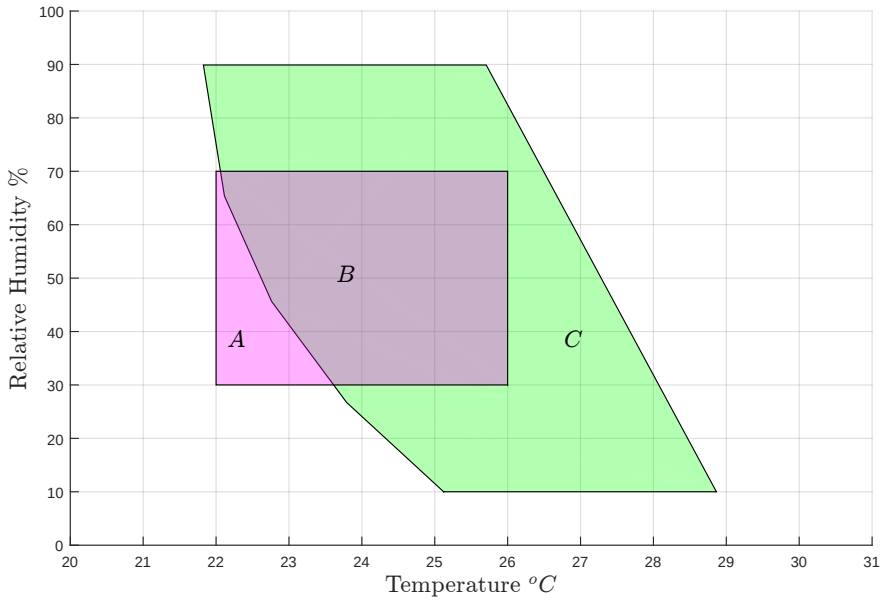


Figure 4.8: Beijing Case: Box comfort zone vs. Learned comfort zone at the end of the day.

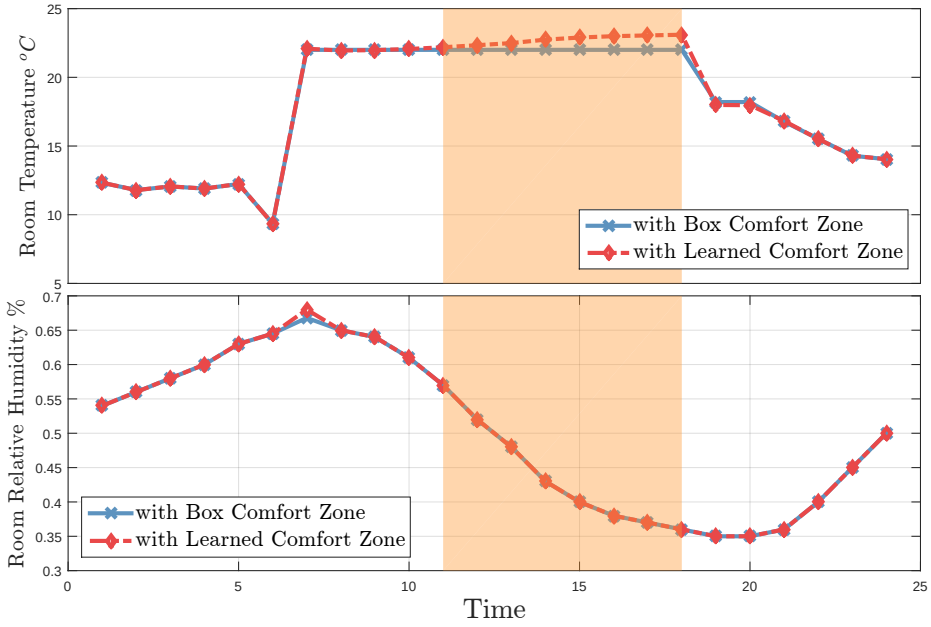


Figure 4.9: Beijing Case: Operated room temperature (top) and relative humidity (bottom) for MPC1 and MPC2..

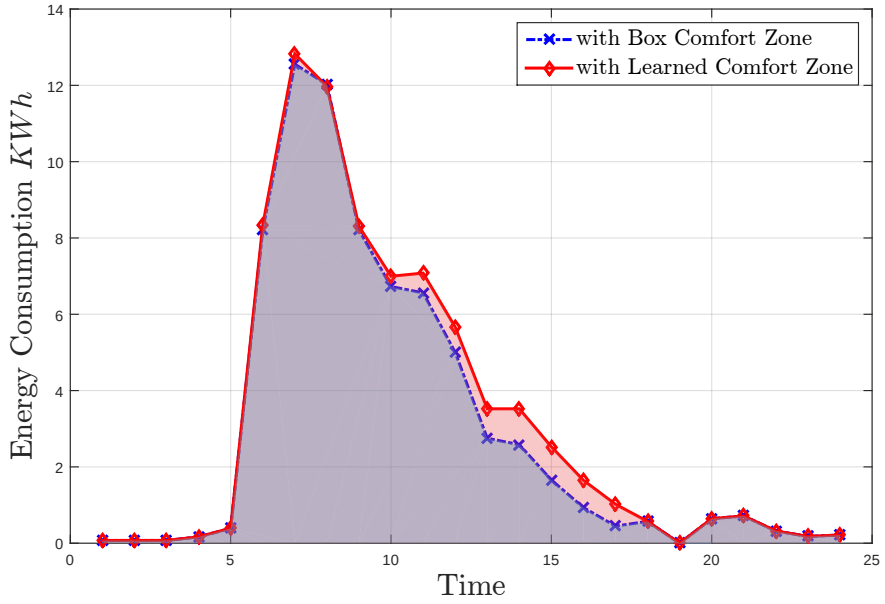


Figure 4.10: Beijing Case: Total HVAC energy usage for MPC1 and MPC2.

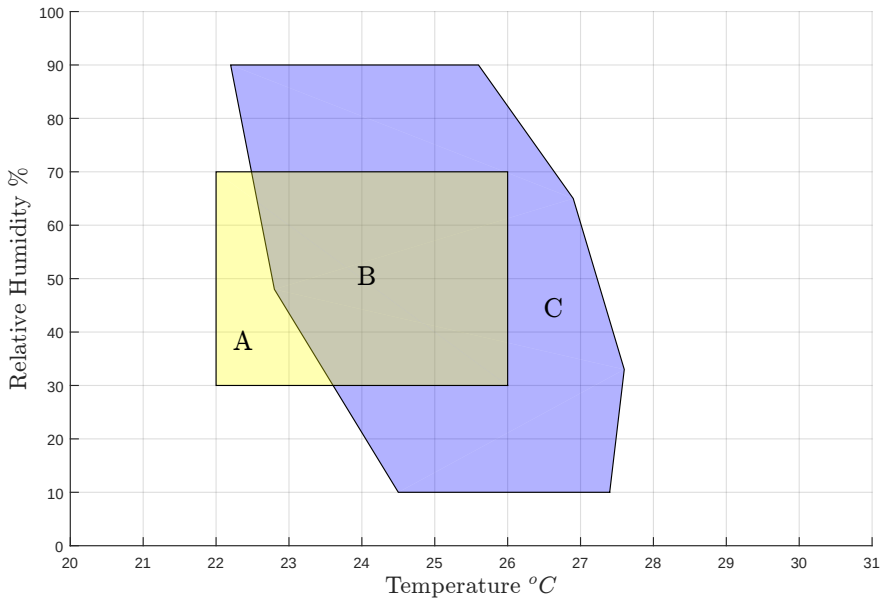


Figure 4.11: Singapore Case: Box comfort zone vs. Learned comfort zone at the end of the day.

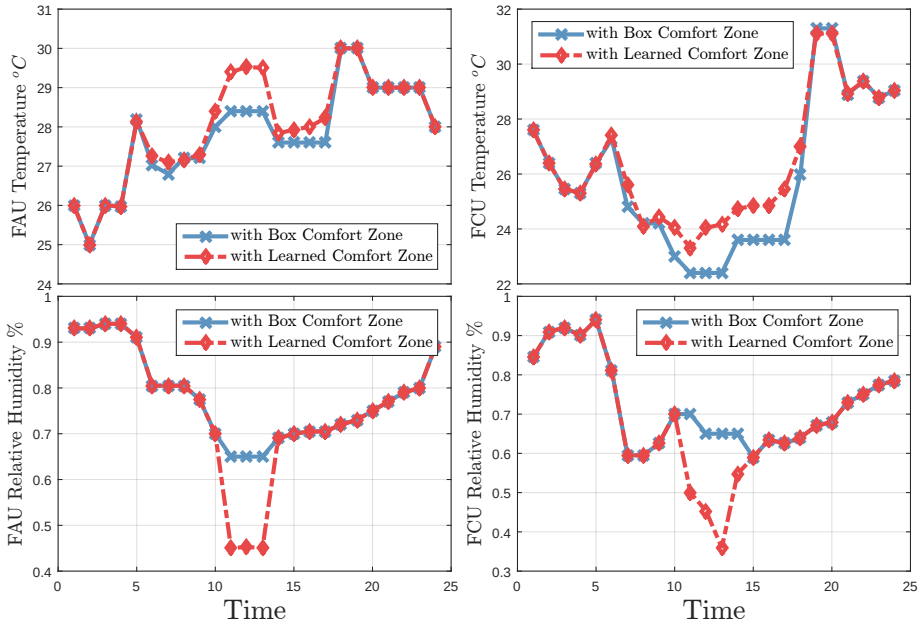


Figure 4.12: Singapore Case: Temperature and Relative Humidity set points for MPC1 and MPC2. Top FAC(left), FCU(right) Temperature; Bottom FAC(left) FCU(right) humidity.

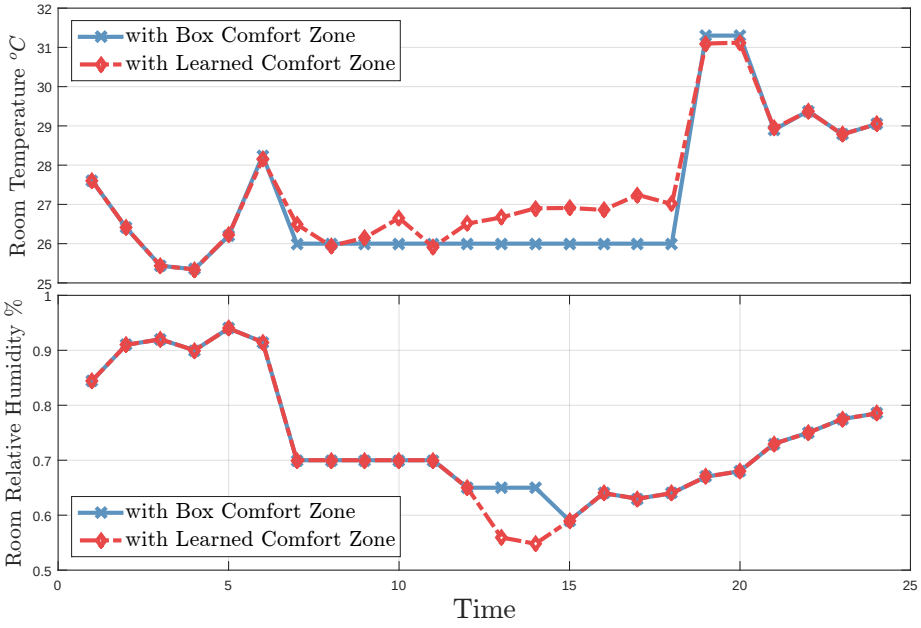


Figure 4.13: Singapore Case: Room temperature (top) and relative humidity (bottom) for MPC1 and MPC2.

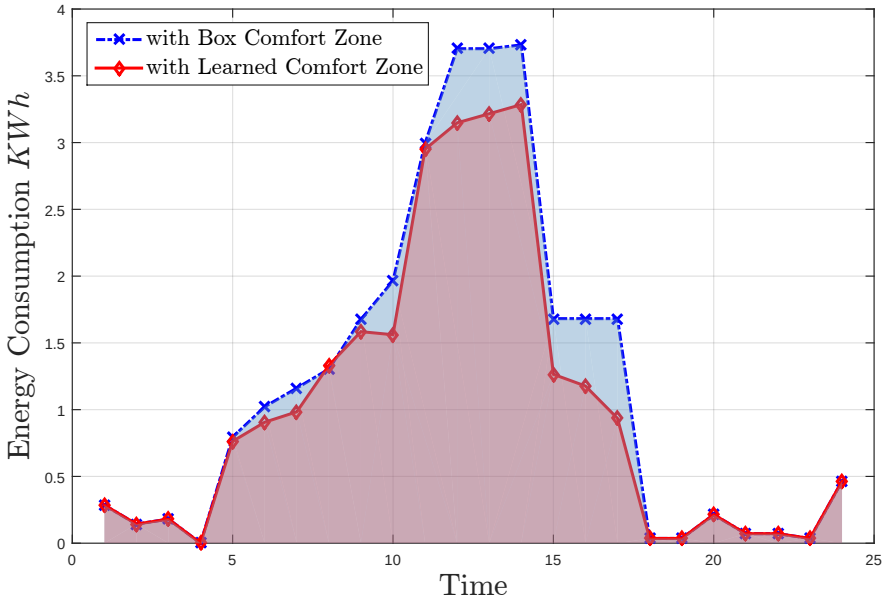


Figure 4.14: Singapore Case: Total HVAC Energy consumption for MPC1 and MPC2.

Condition II: Cooling Modes with Hot and Humid Weather

More interestingly, our second study deals with hot and humid weather condition in Singapore, where the energy efficiency of HVAC system is a critical problem since it is indispensable to regulate indoor air all year long to ensure an acceptable environment. Again 48 hours (0am 18/June to 0am 20/June 2016) weather prediction data for Singapore is used, while the HVAC system is operated in the mode of cooling and dehumidification.

The CPLM based comfort zone learned at the end of the day is shown in Figure 4.11 together with the fixed box comfort model. Since the collected data corresponds to the actually room temperature and humidity values, compared to the Bejing case in Figure 4.8, the right side of the comfort zone is more refined. Once again we see that the box comfort model is too conservative and ignores potential comfort region C (hot but low humidity), which in the current Singapore case serves as the active constraints for the HVAC system MPC. Hence by exploiting the extra feasible region C, one can achieve better HVAC scheduling. The results in Figure 4.12 - Figure 4.14 confirm this intuition.

Figure 4.12 shows the values of control variables, i.e. the temperature and relative humidity set points for FAC and FCU, and Figure 4.13 shows the room temperature and humidity created by this HVAC system scheduling. Interestingly compared to MPC1 (with box comfort zone), in the period from 10am to 3pm, the MPC2 (with CPLM based comfort zone) operates at a much lower room humidity while allowing some raise in the room temperature. Since in the afternoon in Singapore the outdoor temperature is fairly high but humidity is relatively low, for the HVAC operation cooling is rather expensive but dehumidification is

cheap during this period. As a consequence, with a more flexible comfort constraint produced by CPLM (region C in Figure 4.11), the MPC2 is able to save energy usage. The hourly energy consumption curves are given in Figure 4.14 and it is seen that MPC2 uses much less energy from 10am to 5pm. Summing up for 24 hours, in total MPC2 consumes **12.81% less** energy than MPC1 does. The result indicates that simply by applying the CPLM based comfort modeling method for HVAC MPC, one can achieve significant energy saving by exploiting the margins of comfort requirement.

4.4.3 Case Study: PMU based Event Detection

4.4.3.1 Experiment Setup and Feature Engineering

As a continuation of the experiment conducted in Chapter 3, we test the proposed HS^{3}M learning framework and the PDM optimization algorithm for power distribution system event classification and diagnosis, after outliers or novelties have been detected using the methods proposed in Chapter 3. The distribution network setup and the high resolution μPMU data collection procure are the same as introduced in Section 3.4.1. It is worth noting that previous chapter is focused on detection with time series data, hence the models have to include temporal dependence. In this experiment, however, we take the chunk of data marked outliers/novelties as input, and treat each marked window of the data as independent observations of a particular event, and thus temporal dependence is not a concern for this application. In other words, we are interested in identifying events types based on useful information (features) extracted from the outlier window. This is illustrated in Figure 4.15.

Four types of commonly encountered events in power distribution networks, namely Voltage Disturbance (VD) and Voltage Sag (VS), Motor Start (MS), High Impedance fault (HI), are considered in this experiment. For the ease of notation let $w_t^i \triangleq \{x_t^i, \cdots, x_{t+L}^i\}$ be the t^{th} outlier window of measurement i. Since all artificial intelligent methods are "garbage in, garbage out", we consider diverse techniques to construct feature candidates. Intuitively, some events, such as voltage sag or voltage disturbance, could be revealed by investigating single streams (voltage magnitude or phase) fluctuations, while other events, such as high impedance fault and voltage oscillation, might be more obvious by analyzing the inter-behavior/dependence of multiple voltage and current streams. For the purpose of detecting different types of events, we include both single stream and inter-stream feature extract with a variety of metrics. To be specific, we consider

Single-stream Features

- Classic statistics: including mean, variance, and range of voltage/current magnitude in each window. These features capture the average voltage/current values as well as their fluctuations in the time slot. The median is also included as it is a more "robust" metric of average value from a statistical viewpoint. To further characterize the variations of magnitude in each window, the distributional features, including entropy and histogram are calculated.

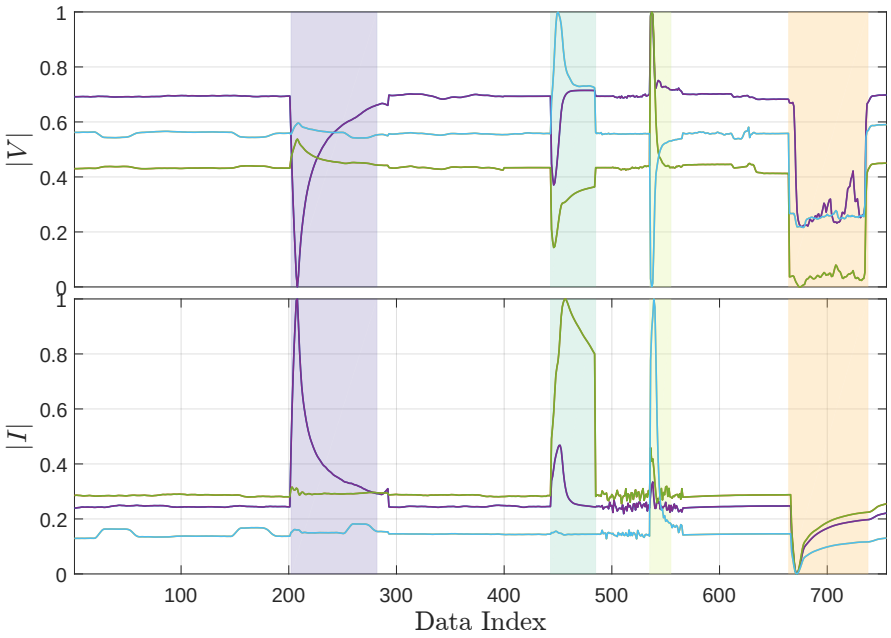


Figure 4.15: Detected window of outliers for further event classification. Note that some periods of stable state are shrunk and the events are zoomed out for visualization purpose.

- First order difference: We compute $x^i_{t+1} - x^i_t$ for each stream and take the corresponding mean and variance in each window. The intuition is that some transient events may exhibit significant "jumps" in voltage and current magnitude, which can be well captured by "spikes" in the first order difference. As for streams associated with phase information, the average difference is an indicator of voltage/current frequency and is also an important indicator of system stability.

- Transformation: Notice that many distribution side events, such as ON/OFF of reactive loads, usually lead to oscillations in both magnitude and phase measurement, we propose to use Fast Fourier Transform (FFT) to capture this frequency domain information. Also, Wavelet transformation is adopted to capture local fluctuations and abrupt changes.

Inter-stream Features

- Deviation: the difference between any two of the three phases, for both voltage and current. The resulted time sequences are processed as single streams in each window with classic statistics. In this way, we incorporate information for the events that exhibit phase imbalance.

- Correlation between any two of the three phases, for both voltage and current. The correlation constitutes a metric of dependence for these time series, and is also helpful in providing information related with inter-phase behavior.

Table 4.5: Extracted Features Candidates

Single Stream	Statistics	$\text{mean}(w_t^i), \text{var}(w_t^i), \text{range}(w_t^i)$ $\text{median}(w_t^i), \text{entropy}(w_t^i), \text{hist}(w_t^i)$
	Difference	$u_t^i = \text{Diff}(x_t^i)$; Statistics
	Transformation	$\text{fft}(w_t^i), \text{wavelet}(w_t^i)$
Inter Stream	Deviation	$x^i - x^j \quad \forall i, \forall j \in \mathcal{N}(i)$
	Correlation	$corr(x^i, x^j) \quad \forall i, \forall j \in \mathcal{N}(i)$

A summary of feature extraction candidates are given in Table 4.5. Note that the inter-stream features for different nodes (hence from different μPMUs) should be very interesting for sub-systems width event detection, for which one can include not only correlation as dependence metric, but also causal information [29] that pinpoints the propagation of the event. The task of identifying sub-system scale events and their influence on neighboring nodes is one of our future work. With the presented feature extraction procedure, a total number of 312 features have been obtained. However, some of them may be redundant as there are significant similarities among extracted features, for example, when the three phases are balanced, their single stream mean, variation, etc., are almost the same. From a machine learning point of view, adding redundant features does not help event detection, but instead introduces extra noise and cause computational difficulties. In this work, we adopt a method developed in [206], called Minimum-redundancy-maximum-relevance (mRMR). The procedure uses mutual information as the metric of goodness of a feature set, and resolve the trade-off between relevancy and redundancy. For each event, mRMR is conducted to choose 50 most informative features. Also note that all numerical experiments in the following are conducted on a workstation having dual Xeon X5687 CPUs and 72GB memory.

4.4.3.2 The Performance of HS^{3}M

The training set contains about 40000 μPMU records with detailed labels (completely labeled data). The testing data set contains the similar events and has around 30000 data points, but is collected at a different time. For the training of HS^{3}M which enables the inclusion of partial knowledge, another 36000 partially labeled data and 108000 unlabeled data are also used (the effect of the size of these data sets will be discussed later).

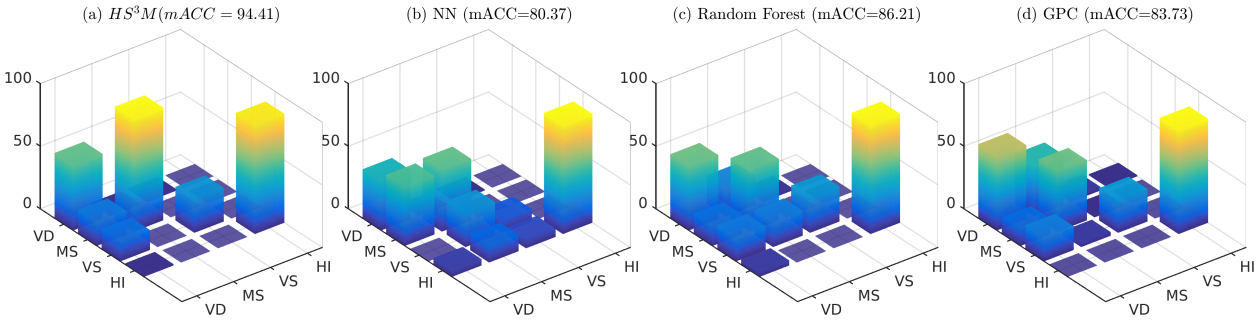


Figure 4.16: Confusion Matrix for different methods. Diagonal terms are correct identifications and off-diagonal ones are mis-classifications. mACC for multi-class detection accuracy. Note that the class of stable state is not included for better visualization.

The performance of HS^{3}M is compared to that of other popular multi-class event detection methods, including a 3-layer neural network (NN), Decision Tree based Random Forest (RF), and Gaussian Process Classification (GPC). The hyper-parameters of those models, e.g., the cost balance coefficient, are chosen with 10-folds cross validation (CV). The confusion matrices (contingency table) for all methods are shown in Figure 4.16. Each row of the sub-figure represents the samples in predicted class while each column represents the samples in actual (true) class. The overall multi-class accuracy (mACC) is summarized in the title of each sub-figure. We see from the confusion matrices that significant improvement is achieved: HS^{3}M provides 94.41% mACC, outperforming the best of the other methods by around 8%, while the classic NN only yields 80% accuracy. Moreover, HS^{3}M gives improvements in differentiating all event types, especially VS, MS, and HI with an accuracy at least 90%. The only issue is that it tends to confuse VD with VS, which is somewhat expected as the criteria for distinguishing VD and VS events are thresholding on the voltage magnitude. In short, the results justified the effectiveness of the proposed HS^{3}M, as well as the idea of incorporating partial information for event detection.

The computational cost in terms of training and testing (or prediction) time/memory usage are also listed in Table 4.6. It appears that HS^{3}M requires a longer time and a median memory usage in the training phase. This is expected since HS^{3}M is a more comprehensive method incorporating partial information. On the other hand, the testing time/memory usage of HS^{3}M are one of the shortest/smallest. This is due to the solution sparsity of the HS^{3}M classifier. In practice, testing cost is of major concern because event prediction should be done in real time on distributed systems, while training can be performed "offline" on powerful computers. In this regards, the proposed HS^{3}M is promising also when computational cost is a concern.

Table 4.6: Comparison of Computational Cost.

Method	HS^{3}M	NN	RF	GPC
Time Train (min)	35.7	17.4	11.8	28.6
Time Test (sec)	53.7	46.6	81.2	221.9
Mem. Train (MB)	193	76	59	299
Mem. Test (MB)	0.79	1.91	0.52	292

4.4.3.3 Effect of Partial Information

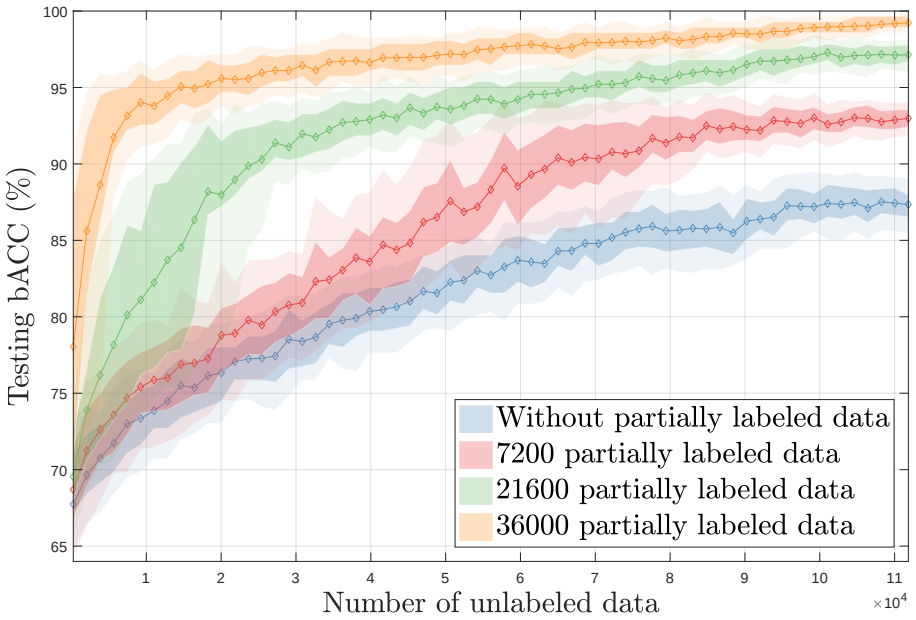


Figure 4.17: The incorporation of partially and unlabeled data

Last but not least, the benefit of including additional partially labeled data and unlabeled data is investigated. To do this, HS^{3}M is evaluated with 0, 7200, 21600 and 36000 partially labeled data samples and 0 - 108000 unlabeled data points. The testing accuracy of each test, averaged over 50 random sampled experiments, is plotted in Figure 4.17 (diamond line), together with the 0.75 and 0.90 confidence intervals (shaded area). Note that when no partially labeled data is included (blue line), HS^{3}M reduces to a version of semi-supervised learning machine. In the case where unlabeled data is not incorporated (the beginning of each line), HS^{3}M can be viewed as VCMKL. In general, it is observed that the performance improves as more partially labeled data and unlabeled data are added, while the improvements exhibit a "diminishing return" property, i.e., the marginal benefit of including more

and more partially/unlabeled data is decreasing. Besides, it appears that HS^{3}M, by leveraging both source of information, significantly outperforms previous semi-supervised and consensus learning methods.

4.5 Appendix: Proofs

4.5.1 Generalization Analysis for the Proposed Classifiers

Lemma. $pM \leq d \leq 2(p+1)M\log_2[(p+1)M]$

Proof. • **lower bound:** Consider a p dimensional hypersphere and M hyperplanes that cut the hypersphere at intersections denoted as $\mathcal{I}_1, \cdots, \mathcal{I}_M$ (d dimensional "circle"). For each $\mathcal{I}_j$, put p points on it, denoted as $\mathcal{X}_1, \cdots, \mathcal{X}_M$, with $|\mathcal{X}_j| = p$.

We know that the p points on each "circle" $\mathcal{I}_j$ could be shattered by the associated hyperplane. In addition, since all points lies on the hypersphere, for every shattering with the hyperplane, we can require that the rest of the points $\cup_{m\neq j}\mathcal{X}_m$ are labeled $+1$. In this way the j^{th} hyperplane only affects the labeling of the points lies on $\mathcal{I}_j$. Thus $|\cup_{m=1}^{M}\mathcal{X}_m| = pM$ points can be arbitrarily labeled by CPLC with M hyperplanes, which gives $d \geq pM$.

• **upper bound:** Let the function class of p dimensional M hyperplanes CPLC as $\mathcal{G}$, and that of p dimensional hyperplane as $\mathcal{H}$. Consider the growth function of l points

$$\Pi_{\mathcal{G}}(l) \triangleq \max\{|\mathcal{G}_{\mathcal{X}}|, |\mathcal{X}| = l\}$$

for $\mathcal{G}$ and similarly $\Pi_{\mathcal{H}}(l)$ for $\mathcal{H}$. We know that the VC dimension of p dimensional hyperplane is just $p+1$, by Sauer's Theorem we have for $l \geq p+1$ and $p \geq 2$

$$\Pi_{\mathcal{H}}(l) \leq \sum_{i=0}^{p+1}\binom{l}{i} \leq \left(\frac{el}{p+1}\right)^{p+1} \leq l^{p+1}$$

Because the CPLC is comprised of M hyperplanes, each of which is able to generate at most l^{p+1} labelings, we have

$$\Pi_{\mathcal{G}}(l) \leq l^{(p+1)M}$$

In addition one can easily check that $l^{(p+1)M} \leq 2^l$ for

$$l = \lceil 2(p+1)M\log_2((p+1)M)\rceil$$

□

Theorem. *Denote* $d' \triangleq 2(p+1)M\log_2[(p+1)M]$, $R(g)$ *the generalization risk (0-1 loss) and* $\widehat{R}(g)$ *the empirical risk. Assume large enough sample size* $l > d'$, *we have that with probability at least* $1-\delta$

$$R(g) \leq \widehat{R}(g) + \sqrt{\frac{2\log(el/d')}{l/d'}} + \sqrt{\frac{\log(1/\delta)}{2l}}$$

Proof. It follows directly by combining the above lemma and the classic VC-dimension generalization bound (see Corollary 3.4 of [147]), and using the fact that the function $\frac{\log(el/d)}{l/d}$ is monotonically increasing in d when $l \geq d$. □

Theorem. *The function class $\mathcal{G}$ of VCMKL has*

$$\widehat{\mathcal{R}}(\mathcal{G}\left(\boldsymbol{x}_1^l\right)) \leq 2\sum_{j=1}^{M} \widehat{\mathcal{R}}\left(\mathcal{F}_j(\boldsymbol{x}_1^l)\right)$$

Further assume $\mathcal{F}_j$ forms a bounded function class with kernel $\kappa_j(\cdot,\cdot)$ and kernel matrix $\boldsymbol{K}_j$ such that $\mathcal{F}_j = \left\{\boldsymbol{x} \mapsto \sum_{i=1}^{l} \alpha_i \kappa_j(\boldsymbol{x}_i, \boldsymbol{x}) \mid \boldsymbol{\alpha}^T \boldsymbol{K}_j \boldsymbol{\alpha} \leq B_j\right\}$ then $\widehat{\mathcal{R}}\left(\mathcal{G}(\boldsymbol{x}_1^l)\right) \leq \frac{4}{l}\sum_{j=1}^{M} B_j \sqrt{tr(\boldsymbol{K}_j)}$.

Proof. For the first part, we need the following lemma

Lemma. *Talagrand's Lemma*
Let $\Phi : \mathbb{R} \mapsto \mathbb{R}$ be $\eta -$ Lipschitz, and $\Upsilon : \mathbb{R} \mapsto \mathbb{R}$ be convex and nondecreasing. Then for any function class $\mathcal{F}$ of real-valued functions, the following inequality holds:

$$\hat{\mathcal{R}}(\Upsilon \circ \Phi \circ \mathcal{F}(\boldsymbol{x}_1^l)) \leq \eta \hat{\mathcal{R}}(\Upsilon \circ \mathcal{F}(\boldsymbol{x}_1^l))$$

Now for the main theorem, consider the case $M = 2$. The following inequality is straightforward:

$$\begin{aligned}
&\sup_{g\in\mathcal{G}} \left| \frac{1}{l}\sum_{i=1}^{l} \sigma_i g(\boldsymbol{x}_i) \right| \\
&\leq \left[\sup_{g\in\mathcal{G}} \frac{1}{l}\sum_{i=1}^{l} \sigma_i g(\boldsymbol{x}_i) \right]_+ + \left[\sup_{g\in\mathcal{G}} \frac{1}{l}\sum_{i=1}^{l} -\sigma_i g(\boldsymbol{x}_i) \right]_+
\end{aligned}$$

Noticing that $-\sigma_1, \cdots, -\sigma_l$ has the same distribution as $\sigma_1, \cdots, \sigma_l$, we get

$$\begin{aligned}
\hat{\mathcal{R}}\left(\mathcal{G}(\boldsymbol{x}_1^l)\right) &= E_{\boldsymbol{\sigma}}\left[\sup_{g\in\mathcal{G}} \left| \frac{1}{l}\sum_{i=1}^{l} \sigma_i g(\boldsymbol{x}_i) \right| \right] \\
&\leq 2E_{\boldsymbol{\sigma}}\left[\sup_{g\in\mathcal{G}} \frac{1}{l}\sum_{i=1}^{l} \sigma_i g(\boldsymbol{x}_i) \right]_+
\end{aligned}$$

Writing $g = \min\{f_1, f_2\} = \frac{1}{2}(f_1 + f_2) - \frac{1}{2}|f_1 - f_2|$, the last term yields

$$\begin{aligned}
&\left[\sup_{g\in\mathcal{G}} \frac{1}{l}\sum_{i=1}^{l} \sigma_i g(\boldsymbol{x}_i)\right]_+ \\
&\overset{(a)}{\leq} \left[\sup_{\mathcal{F}_1,\mathcal{F}_1} \frac{1}{l}\sum_{i=1}^{l} \sigma_i \frac{1}{2}(f_1(\boldsymbol{x}_i) + f_2(\boldsymbol{x}_i))\right]_+ \\
&\quad + \left[\sup_{\mathcal{F}_1,\mathcal{F}_1} \frac{1}{l}\sum_{i=1}^{l} -\sigma_i \frac{1}{2}|f_1(\boldsymbol{x}_i) - f_2(\boldsymbol{x}_i)|\right]_+ \\
&\overset{(b)}{\leq} \frac{1}{2}\left[\sup_{\mathcal{F}_1} \frac{1}{l}\sum_{i=1}^{l} \sigma_i f_1(\boldsymbol{x}_i)\right]_+ + \frac{1}{2}\left[\sup_{\mathcal{F}_2} \frac{1}{l}\sum_{i=1}^{l} \sigma_i f_2(\boldsymbol{x}_i)\right]_+ \\
&\quad + \left[\sup_{\mathcal{F}_1,\mathcal{F}_1} \frac{1}{l}\sum_{i=1}^{l} -\sigma_i \frac{1}{2}|f_1(\boldsymbol{x}_i) - f_2(\boldsymbol{x}_i)|\right]_+
\end{aligned}$$

where (a) and (b) are due to the upper additive property of sup and $[\cdot] = max\{0, \cdot\}$ function. Taking expectations for this upper bound and applying Talagrand's Lemma with $\Upsilon = [\cdot]$ and $\Phi = |\cdot|$ yields

$$\begin{aligned}
&E_{\boldsymbol{\sigma}}\left[\sup_{\mathcal{F}_1,\mathcal{F}_1} \frac{1}{l}\sum_{i=1}^{l} -\sigma_i \frac{1}{2}|f_1(\boldsymbol{x}_i) - f_2(\boldsymbol{x}_i)|\right]_+ \\
&\leq \frac{1}{2}E_{\boldsymbol{\sigma}}\left[\sup_{\mathcal{F}_1} \frac{1}{l}\sum_{i=1}^{l} \sigma_i f_1(\boldsymbol{x}_i)\right]_+ + \frac{1}{2}E_{\boldsymbol{\sigma}}\left[\sup_{\mathcal{F}_2} \frac{1}{l}\sum_{i=1}^{l} \sigma_i f_2(\boldsymbol{x}_i)\right]_+
\end{aligned}$$

Putting two inequalities together we have

$$\begin{aligned}
&E_{\boldsymbol{\sigma}}\left[\sup_{g\in\mathcal{G}} \frac{1}{l}\sum_{i=1}^{l} \sigma_i g(\boldsymbol{x}_i)\right]_+ \\
&\leq \frac{1}{2}E_{\boldsymbol{\sigma}}\left[\sup_{\mathcal{F}_1} \frac{1}{l}\sum_{i=1}^{l} \sigma_i f_1(\boldsymbol{x}_i)\right]_+ + \frac{1}{2}E_{\boldsymbol{\sigma}}\left[\sup_{\mathcal{F}_2} \frac{1}{l}\sum_{i=1}^{l} \sigma_i f_2(\boldsymbol{x}_i)\right]_+ \\
&\quad + \frac{1}{2}E_{\boldsymbol{\sigma}}\left[\sup_{\mathcal{F}_1} \frac{1}{l}\sum_{i=1}^{l} \sigma_i f_1(\boldsymbol{x}_i)\right]_+ + \frac{1}{2}E_{\boldsymbol{\sigma}}\left[\sup_{\mathcal{F}_2} \frac{1}{l}\sum_{i=1}^{l} \sigma_i f_2(\boldsymbol{x}_i)\right]_+ \\
&\leq E_{\boldsymbol{\sigma}}\left[\sup_{\mathcal{F}_1} \left|\frac{1}{l}\sum_{i=1}^{l} \sigma_i f_1(\boldsymbol{x}_i)\right|\right] + E_{\boldsymbol{\sigma}}\left[\sup_{\mathcal{F}_2} \left|\frac{1}{l}\sum_{i=1}^{l} \sigma_i f_2(\boldsymbol{x}_i)\right|\right] \\
&= \hat{\mathcal{R}}\left(\mathcal{F}_1(\boldsymbol{x}_1^l)\right) + \hat{\mathcal{R}}\left(\mathcal{F}_2(\boldsymbol{x}_1^l)\right)
\end{aligned}$$

hence finally,

$$\hat{\mathcal{R}}\left(\mathcal{G}(\boldsymbol{x}_1^l)\right) \leq 2\left[\hat{\mathcal{R}}\left(\mathcal{F}_1(\boldsymbol{x}_1^l)\right) + \hat{\mathcal{R}}\left(\mathcal{F}_2(\boldsymbol{x}_1^l)\right)\right]$$

Also it is straightforward to generalize the above argument to $M > 2$ with simple induction. Finally we get

$$\hat{\mathcal{R}}(\mathcal{G}\left(\boldsymbol{x}_1^l\right)) \leq 2\sum_{j=1}^{M} \hat{\mathcal{R}}\left(\mathcal{F}_j(\boldsymbol{x}_1^l)\right)$$

The second part of the theorem can be obtained with a standard approach in bounding empirical Rademacher complexity of Kernels. □

4.5.2 Lemma and Theorems for PDM: Reformulation

For ease of notation, the inner minimization is repeated here:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \mathcal{J}(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{Q}\boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\alpha} \\ \text{subject to} \quad & \boldsymbol{C}^\alpha\boldsymbol{\alpha} \leq \boldsymbol{C}^p\boldsymbol{p} + \boldsymbol{C}^0, \quad \boldsymbol{y}^T\boldsymbol{\alpha} = 0. \end{aligned} \tag{IO}$$

Lemma. *If the solution $\boldsymbol{\alpha}^*$ of (IO) is non-degenerate, then*

- *The matrix $\boldsymbol{H} \triangleq \frac{\boldsymbol{Q}^{-1}\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{Q}^{-1}}{\boldsymbol{y}^T\boldsymbol{Q}^{-1}\boldsymbol{y}} - \boldsymbol{Q}^{-1}$ is negative semidefinite, and $\boldsymbol{R} \triangleq \boldsymbol{C}_{\mathcal{A}}^{\alpha}\boldsymbol{H}\boldsymbol{C}_{\mathcal{A}}^{\alpha T}$ is strictly negative definite, hence is invertible.*
- *The parametric solution $\boldsymbol{\alpha}^*(\boldsymbol{p})$ exists and is unique.*

Proof. We first prove the following claim:

Claim. *If the solution $\boldsymbol{\alpha}^*$ is Non-Degenerate, the over-determined system for the variable $\boldsymbol{\xi}$*

$$\left(\boldsymbol{C}_{\mathcal{A}}^{\alpha}\right)^T\boldsymbol{\xi} = \boldsymbol{y} \tag{4.37}$$

does not have a solution.

Note that the rows of $\boldsymbol{C}^\alpha$ only contains n dimensional standard basis $\{\pm e_1^T, \cdots, \pm e_N^T\}$. The active constraints are induced by non-support vectors ($\alpha_i^* = 0$) in the first block, and bounded support vectors ($\alpha_i^* = c_i p_i$) in the second block. These active constraints must be orthogonal and span a subspace of $\mathbb{R}^n$, because α_i^* cannot be 0 and $c_i p_i$ at the same time. For any $k \in \mathcal{A}^c$, the basis $\pm e_k$ are not in $\boldsymbol{C}_{\mathcal{A}}^{\alpha}$, hence the k_{th} column vectors of $\boldsymbol{C}_{\mathcal{A}}^{\alpha}$ must be all zero. When the solution is non-degenerate, there exist at least two indexes in $\mathcal{A}^C$, with one corresponding to $y_k = 1$ and another corresponding to $y_{k'} = -1$, which results in conflict equations $0 = -1$ or $0 = +1$. Thus $(\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T\boldsymbol{\xi} = \boldsymbol{y}$ is an inconsistent system with no solution.

Then we claim that:

Claim. *$\boldsymbol{H}$ has rank $N-1$, and is a symmetric negative semi-definite matrix.*

By the Rank-nullity theorem, we know that $\text{rank}(\boldsymbol{H}) + \text{nul}(\boldsymbol{H}) = N$. We now consider the nullspace of $\boldsymbol{H}$. First, note that if a vector $\boldsymbol{v} \in \mathbb{R}^N$ is in the nullspace of $\boldsymbol{H}$,

$$\boldsymbol{H}\boldsymbol{v} = 0 \Longleftrightarrow (\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{Q}^{-1})\boldsymbol{v} = (\boldsymbol{y}^T\boldsymbol{Q}^{-1}\boldsymbol{y})\boldsymbol{v}. \tag{4.38}$$

by the definition of $\boldsymbol{H}$. Since $\text{rank}(\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{Q}^{-1}) = 1$, this means $\boldsymbol{v}$ is the eigenvector corresponding to the only non-zero eigenvalue of $\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{Q}^{-1}$. It is straightforward to check $\boldsymbol{v} = \boldsymbol{y}$ and hence $\text{null}(\boldsymbol{H}) = \text{span}\{\boldsymbol{y}\}$, proving that $\text{rank}(\boldsymbol{H}) = N - 1$.

Now we look at $\boldsymbol{v}^T\boldsymbol{H}\boldsymbol{v} \;\; \forall \boldsymbol{v} \in \mathbb{R}^N$.

$$\begin{aligned} \boldsymbol{v}^T\boldsymbol{H}\boldsymbol{v} &= \boldsymbol{v}^T \left(\frac{\boldsymbol{Q}^{-1}\boldsymbol{y}\boldsymbol{y}^T\boldsymbol{Q}^{-1}}{\boldsymbol{y}^T\boldsymbol{Q}^{-1}\boldsymbol{y}} - \boldsymbol{Q}^{-1} \right) \boldsymbol{v} \\ &= \frac{(\boldsymbol{v}^T\boldsymbol{Q}^{-1}\boldsymbol{y})^2 - (\boldsymbol{v}^T\boldsymbol{Q}^{-1}\boldsymbol{v})(\boldsymbol{y}^T\boldsymbol{Q}^{-1}\boldsymbol{y})}{\boldsymbol{y}^T\boldsymbol{Q}^{-1}\boldsymbol{y}}. \end{aligned} \tag{4.39}$$

Since $\boldsymbol{Q}^{-1}$ is a positive definite matrix, it has a Cholesky decomposition $\boldsymbol{Q}^{-1} = \boldsymbol{L}\boldsymbol{L}^T$ where $\boldsymbol{L}$ is a lower triangular matrix. Defining $\tilde{\boldsymbol{v}} \triangleq \boldsymbol{L}^T\boldsymbol{v}$, and $\tilde{\boldsymbol{y}} \triangleq \boldsymbol{L}^T\boldsymbol{y}$ the numerator becomes

$$\begin{aligned} &(\boldsymbol{v}^T\boldsymbol{Q}^{-1}\boldsymbol{y})^2 - (\boldsymbol{v}^T\boldsymbol{Q}^{-1}\boldsymbol{v})(\boldsymbol{y}^T\boldsymbol{Q}^{-1}\boldsymbol{y}) \\ &=(\boldsymbol{v}^T\boldsymbol{L}\boldsymbol{L}^T\boldsymbol{y})^2 - (\boldsymbol{v}^T\boldsymbol{L}\boldsymbol{L}^T\boldsymbol{z})(\boldsymbol{y}^T\boldsymbol{L}\boldsymbol{L}^T\boldsymbol{y}) \\ &=|\langle\tilde{\boldsymbol{v}}, \tilde{\boldsymbol{y}}\rangle|^2 - ||\tilde{\boldsymbol{v}}||^2||\tilde{\boldsymbol{y}}||^2. \end{aligned} \tag{4.40}$$

By the Cauchy-Schwarz inequality, $|\langle\tilde{\boldsymbol{v}}, \tilde{\boldsymbol{y}}\rangle|^2 \leq ||\tilde{\boldsymbol{v}}||^2||\tilde{\boldsymbol{y}}||^2$. Thus the numerator of $\boldsymbol{v}^T\boldsymbol{H}\boldsymbol{v}$ is ≤ 0. Therefore, $\boldsymbol{H}$ is a negative semi-definite matrix.

Based on that, we have

Claim. *$\boldsymbol{R}$ is a symmetric strictly negative definite matrix.*

Since $\boldsymbol{R} = \boldsymbol{C}_{\mathcal{A}}^{\alpha}\boldsymbol{H}(\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T$, we look at $-\boldsymbol{\xi}^T\boldsymbol{R}\boldsymbol{\xi} \;\; \forall \boldsymbol{\xi} \in \mathbb{R}^N$:

$$\begin{aligned} -\boldsymbol{\xi}^T\boldsymbol{C}_{\mathcal{A}}^{\alpha}\boldsymbol{H}\boldsymbol{C}_{\mathcal{A}}^{\alpha T}\boldsymbol{\xi} &= \boldsymbol{\xi}^T\boldsymbol{C}_{\mathcal{A}}^{\alpha}\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{C}_{\mathcal{A}}^{\alpha T}\boldsymbol{\xi} \\ &= \boldsymbol{\xi}^T\boldsymbol{C}_{\mathcal{A}}^{\alpha}\boldsymbol{U}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{U}^T\boldsymbol{C}_{\mathcal{A}}^{\alpha T}\boldsymbol{\xi} \\ &= \boldsymbol{\xi}^T\boldsymbol{C}_{\mathcal{A}}^{\alpha}\boldsymbol{P}\boldsymbol{P}^T\boldsymbol{C}_{\mathcal{A}}^{\alpha T}\boldsymbol{\xi} \\ &= \tilde{\boldsymbol{\xi}}^T\tilde{\boldsymbol{\xi}} \geq 0 \end{aligned} \tag{4.41}$$

where we've used the spectral theorem to decompose $-\boldsymbol{H}$ and defined $\tilde{\boldsymbol{\xi}} \triangleq \boldsymbol{P}^T(\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T\boldsymbol{\xi}$. If $\exists\tilde{\boldsymbol{\xi}} \mid \tilde{\boldsymbol{\xi}}^T\tilde{\boldsymbol{\xi}} = 0$, this implies $\boldsymbol{P}^T(\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T\boldsymbol{\xi} = 0$. Notice that $\boldsymbol{P}$ has the same rank and eigenvector space as $\boldsymbol{H}$, thus, in order for $\boldsymbol{\xi}^T\boldsymbol{P}\boldsymbol{\xi}$ to be 0, we need $(\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T\boldsymbol{\xi} = \boldsymbol{y}$, and by Claim 1.1 this does not have a solution. Therefore, $\boldsymbol{R}$ is a negative definite matrix, and thus is invertible.

We can further show that

Claim. *With non-degeneracy, the matrix $\boldsymbol{E} \triangleq \boldsymbol{H}(\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T\boldsymbol{R}^{-1}\boldsymbol{C}_{\mathcal{A}}^{\alpha}\boldsymbol{H} - \boldsymbol{H}$ is a symmetric strictly negative definite matrix.*

Note that

$$\boldsymbol{E} = \boldsymbol{H}(\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T \left(\boldsymbol{C}_{\mathcal{A}}^{\alpha}H(\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T\right)^{-1}\boldsymbol{C}_{\mathcal{A}}^{\alpha}\boldsymbol{H} - \boldsymbol{H} \tag{4.42}$$

Using Schur's Complement Theorem

$$\boldsymbol{E} \prec 0 \iff \begin{bmatrix} \boldsymbol{C}_{\mathcal{A}}^{\alpha}\boldsymbol{H}(\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T & \boldsymbol{C}_{\mathcal{A}}^{\alpha}\boldsymbol{H} \\ \boldsymbol{H}(\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T & \boldsymbol{H} \end{bmatrix} \prec 0 \tag{4.43}$$

which is equivalent to

$$[\boldsymbol{\eta}^T \ \boldsymbol{\xi}^T] \begin{bmatrix} \boldsymbol{C}_{\mathcal{A}}^{\alpha} \boldsymbol{H} (\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T & \boldsymbol{C}_{\mathcal{A}}^{\alpha} \boldsymbol{H} \\ \boldsymbol{H} (\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T & \boldsymbol{H} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{\xi} \end{bmatrix} < 0 \tag{4.44}$$

$$\iff \quad \boldsymbol{\eta}^T \boldsymbol{C}_{\mathcal{A}}^{\alpha} \boldsymbol{H} (\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T \boldsymbol{\eta} + \boldsymbol{\eta}^T \boldsymbol{C}_{\mathcal{A}}^{\alpha} \boldsymbol{H} \boldsymbol{\xi} + \boldsymbol{\xi}^T \boldsymbol{H} (\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T \boldsymbol{\eta} + \boldsymbol{\xi}^T \boldsymbol{H} \boldsymbol{\xi} < 0$$

This is equivalent to

$$\left(\boldsymbol{\xi}^T + \boldsymbol{\eta}^T \boldsymbol{C}_{\mathcal{A}}^{\alpha}\right) \boldsymbol{H} \left(\boldsymbol{\xi} + (\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T \boldsymbol{\eta}\right) < 0 \tag{4.45}$$

Because $\text{null}(\boldsymbol{H}) = \text{span}\{\boldsymbol{y}\}$, similarly as before we only have to show $\boldsymbol{\xi} + (\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T \boldsymbol{\eta} = c\boldsymbol{y}$ does not have a solution. Again, with non-degeneracy, this can be checked by a similar argument as in the first lemma.

With these invertibility results at hand, the existence and uniqueness of the parametric solution can be obtained directly by solving the parametric program and noting that the Lagrangien multipliers are unique. See the proof of the following theorem.

□

Theorem. *Assume that the solution of (IO) is non-degenerate and induces a set of active and inactive constraints $\mathcal{A}$ and $\mathcal{A}^{\mathcal{C}}$, respectively. With $\boldsymbol{H}$, $\boldsymbol{R}$ defined previously and $\boldsymbol{T} \triangleq \boldsymbol{H}(\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T$, $\widetilde{\boldsymbol{e}} \triangleq \boldsymbol{C}_{\mathcal{A}}^{\alpha} \boldsymbol{H} \boldsymbol{1}$, we have*
(1) The optimal solution is a continuous piecewise affine function of $\boldsymbol{p}$. And in the critical region defined by

$$\begin{cases} \boldsymbol{R}^{-1}(\boldsymbol{C}_{\mathcal{A}}^{p} \boldsymbol{p} + \boldsymbol{C}_{\mathcal{A}}^{0} + \widetilde{\boldsymbol{e}}) \geq 0 \\ \boldsymbol{C}_{\mathcal{A}^C}^{p} \boldsymbol{p} + \boldsymbol{C}_{\mathcal{A}^C}^{0} - \boldsymbol{C}_{\mathcal{A}^C}^{\alpha} \boldsymbol{T} \boldsymbol{R}^{-1}(\boldsymbol{C}_{\mathcal{A}}^{p} \boldsymbol{p} + \boldsymbol{C}_{\mathcal{A}}^{0} + \widetilde{\boldsymbol{e}}) \geq 0 \end{cases} \tag{4.46}$$

the optimal solution $\boldsymbol{\alpha}^*$ of (IO) admits a closed form

$$\boldsymbol{\alpha}^*(\boldsymbol{p}) = \boldsymbol{T} \boldsymbol{R}^{-1}(\boldsymbol{C}_{\mathcal{A}}^{p} \boldsymbol{p} + \boldsymbol{C}_{\mathcal{A}}^{0} + \widetilde{\boldsymbol{e}}) \tag{4.47}$$

*(2) The optimal objective $\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{p}))$ is a **continuous** piece-wise quadratic (**PWQ**) function of $\boldsymbol{p}$.*

Proof. The (IO) problem is equivalent to

$$\begin{aligned} \max_{\zeta, \boldsymbol{\mu}} \min_{\boldsymbol{\alpha}} \quad & \mathcal{L}(\boldsymbol{\alpha}, \zeta, \boldsymbol{\mu}) = \frac{1}{2} \boldsymbol{\alpha}^T \boldsymbol{Q} \boldsymbol{\alpha} - \boldsymbol{1}^T \boldsymbol{\alpha} \\ & \qquad + \zeta(\boldsymbol{\alpha}^T \boldsymbol{y}) + (\boldsymbol{C}^{\alpha} \boldsymbol{\alpha} - \boldsymbol{C}^{\alpha} \boldsymbol{\alpha} - \boldsymbol{C}^{0})^T \boldsymbol{\mu} \\ \text{subject to} \quad & \boldsymbol{\mu} \geq 0. \end{aligned} \tag{4.48}$$

where $\zeta, \boldsymbol{\mu}$ are the Lagrangian multipliers for the equality and inequality constraints, respectively.

The KKT conditions specify for optimal $(\boldsymbol{\alpha}^*, \zeta^*, \boldsymbol{\mu}^*)$,

$$\nabla_\alpha \mathcal{L} = \boldsymbol{Q}\boldsymbol{\alpha}^* + \zeta^* \widetilde{\boldsymbol{y}} + (\boldsymbol{C}^\alpha)^T \boldsymbol{\mu}^* - \mathbf{1} = 0 \tag{4.49a}$$

$$\boldsymbol{y}^T \boldsymbol{\alpha}^* = 0 \tag{4.49b}$$

$$\mu_i^* \geq 0 \qquad \text{for } i = 1, \ldots, n \tag{4.49c}$$

$$\mu_i^* (\boldsymbol{C}^\alpha \boldsymbol{\alpha}^* - \boldsymbol{C}^p \boldsymbol{p} - \boldsymbol{C}^0)_i = 0 \qquad \text{for } i = 1, \ldots, n \tag{4.49d}$$

$$(\boldsymbol{C}^\alpha \boldsymbol{\alpha}^* - \boldsymbol{C}^p \boldsymbol{p} - \boldsymbol{C}^0)_i \leq 0 \qquad \text{for } i = 1, \ldots, n. \tag{4.49e}$$

From (4.49a),

$$\boldsymbol{\alpha}^* = -\boldsymbol{Q}^{-1}(\zeta^* \boldsymbol{y} + (\boldsymbol{C}^\alpha)^T \boldsymbol{\mu}^* - \mathbf{1}). \tag{4.50}$$

where the PD property of a Mercer kernel guarantees the invertibility of $\boldsymbol{Q}$. Now plugging in the above expression for $\boldsymbol{\alpha}^*$ into (4.49b),

$$\begin{aligned} \boldsymbol{y}^T \boldsymbol{\alpha}^* &= -\boldsymbol{y}^T \boldsymbol{Q}^{-1}(\zeta^* \boldsymbol{y} + (\boldsymbol{C}^\alpha)^T \boldsymbol{\mu}^* - \mathbf{1}) = 0 \\ \Rightarrow \zeta^* &= -\frac{\boldsymbol{y}^T \boldsymbol{Q}^{-1} (\boldsymbol{C}^\alpha)^T \boldsymbol{\mu}^*}{\boldsymbol{y}^T \boldsymbol{Q}^{-1} \widetilde{\boldsymbol{y}}} + \frac{\boldsymbol{y}^T \boldsymbol{Q}^{-1}}{\boldsymbol{y}^T \boldsymbol{Q}^{-1} \boldsymbol{y}} \mathbf{1}. \end{aligned} \tag{4.51}$$

There are two cases for the inequality constraints.

$$\begin{cases} \boldsymbol{C}_i^\alpha \boldsymbol{\alpha}^* - \boldsymbol{C}_i^p \boldsymbol{p} - \boldsymbol{C}_i^0 = 0 & \text{for } i \in \mathcal{A} \\ \mu_i^* = 0,\ \boldsymbol{C}_i^\alpha \boldsymbol{\alpha}^* - \boldsymbol{C}_i^p p - \boldsymbol{C}_i^0 < 0 & \text{for } i \in \mathcal{A}^C \end{cases} \tag{4.52}$$

$\boldsymbol{\mu}_\mathcal{A}$ and $\boldsymbol{\mu}_{\mathcal{A}^C}$ similarly represent the elements of $\boldsymbol{\mu}$ for $i \in \mathcal{A}$ and $i \in \mathcal{A}^C$, respectively. Since $(\boldsymbol{C}^\alpha)^T \boldsymbol{\mu}^* = (\boldsymbol{C}_\mathcal{A}^\alpha)^T \boldsymbol{\mu}_\mathcal{A}^*$, the formula for $\boldsymbol{\alpha}^*, \zeta^*$ in equations (4.50) and (4.51) reduces to

$$\boldsymbol{\alpha}^* = -\boldsymbol{Q}^{-1} \left(\zeta^* \boldsymbol{y} + (\boldsymbol{C}_\mathcal{A}^\alpha)^T \boldsymbol{\mu}_\mathcal{A}^* - \mathbf{1} \right) \tag{4.53}$$

$$\zeta^* = -\frac{\boldsymbol{y}^T \boldsymbol{Q}^{-1} (\boldsymbol{C}_\mathcal{A}^\alpha)^T \boldsymbol{\mu}_\mathcal{A}^*}{\boldsymbol{y}^T \boldsymbol{Q}^{-1} \boldsymbol{y}} + \frac{\boldsymbol{y}^T \boldsymbol{Q}^{-1}}{\boldsymbol{y}^T \boldsymbol{Q}^{-1} \boldsymbol{y}} \mathbf{1}. \tag{4.54}$$

Substituting (4.54) into (4.53), we get

$$\boldsymbol{\alpha}^* = \boldsymbol{T} \boldsymbol{\mu}_\mathcal{A}^* - \boldsymbol{H}\mathbf{1}. \tag{4.55}$$

Another equality we get from the active constraints is

$$\boldsymbol{C}_\mathcal{A}^\alpha \boldsymbol{\alpha}^* = \boldsymbol{C}_\mathcal{A}^p \boldsymbol{p} + \boldsymbol{C}_\mathcal{A}^0. \tag{4.56}$$

Combining (4.55) and (4.56), we get an expression for $\boldsymbol{\mu}_\mathcal{A}^*$ since

$$\begin{aligned} \boldsymbol{C}_\mathcal{A}^\alpha (\boldsymbol{T} \boldsymbol{\mu}_\mathcal{A}^* - \boldsymbol{H}\mathbf{1}) &= \boldsymbol{C}_\mathcal{A}^p \boldsymbol{p} + \boldsymbol{C}_\mathcal{A}^0 \\ \Rightarrow \boldsymbol{R} \boldsymbol{\mu}_\mathcal{A}^* - \widetilde{\boldsymbol{e}} &= \boldsymbol{C}_\mathcal{A}^p \boldsymbol{p} + \boldsymbol{C}_\mathcal{A}^0 \\ \Rightarrow \boldsymbol{\mu}_\mathcal{A}^* &= \boldsymbol{R}^{-1} (\boldsymbol{C}_\mathcal{A}^p \boldsymbol{p} + \boldsymbol{C}_\mathcal{A}^0 + \widetilde{\boldsymbol{e}}) \end{aligned} \tag{4.57}$$

with $\boldsymbol{R}$ defined as before, whose invertibility has already been proved. Finally, we get

$$\boldsymbol{\alpha}^*(\boldsymbol{p}) = \boldsymbol{T}\boldsymbol{R}^{-1}(\boldsymbol{C}_{\mathcal{A}}^p\boldsymbol{p} + \boldsymbol{C}_{\mathcal{A}}^0 + \widetilde{\boldsymbol{e}}) - \boldsymbol{H}\boldsymbol{1} \tag{4.58}$$

as an explicit function of $\boldsymbol{p}$.

We now derive the boundaries of the critical region in which (4.58) holds. For active constraints, (4.49c) and (4.49e) require that

$$\boldsymbol{\mu}_{\mathcal{A}}^* \geq 0 \tag{4.59}$$

$$\boldsymbol{C}_{\mathcal{A}^c}^{\alpha}\boldsymbol{\alpha}^* \leq \boldsymbol{C}_{\mathcal{A}^c}^p\boldsymbol{p} + \boldsymbol{C}_{\mathcal{A}^c}^0. \tag{4.60}$$

These two inequalities yield

$$\begin{aligned} &\boldsymbol{R}^{-1}(\boldsymbol{C}_{\mathcal{A}}^p\boldsymbol{p} + \boldsymbol{C}_{\mathcal{A}}^0 + \widetilde{\boldsymbol{e}}) \geq 0 \\ &\boldsymbol{C}_{\mathcal{A}^c}^{\alpha}\boldsymbol{T}\boldsymbol{R}^{-1}(\boldsymbol{C}_{\mathcal{A}}^p\boldsymbol{p} + \boldsymbol{C}_{\mathcal{A}}^0 + \widetilde{\boldsymbol{e}}) \leq \boldsymbol{C}_{\mathcal{A}^c}^p\boldsymbol{p} + \boldsymbol{C}_{\mathcal{A}^c}^0. \end{aligned} \tag{4.61}$$

The proof of continuity relies on the strict convexity of the dual. Since the boundary of any two regions belongs to both closures, and the optimum is unique for all hyperparameters in the feasible set, the solution across the boundary is continuous. With part 2 at hand, part 3 is immediate. □

Theorem. *Still sssuming non-degeneracy, then*

1. *There are finite number of polyhedron critical regions $CR_1, \cdots, CR_{N_r}$ which constitute a **partition** of the feasible set of $\boldsymbol{p}$, i.e. each feasible $\boldsymbol{p}$ belongs to one and only one critical region.*

2. *The optimal objective $\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{p}))$ is a **globally convex** Piece-wise Quadratic (PWQ) function of $\boldsymbol{p}$, and is almost everywhere differentiable.*

3. *The optimal objective is difference-definite, i.e., the differences between its expressions on neighboring polyhedron critical regions have positive or negative semidefinite Hessian.*

4. *Let the common boundary of any two neibouring critical regions CR_i and CR_j be $\boldsymbol{a}^T \backslash p + b$, then there exist a scalar β and a constant c, such that*

$$J_i(\boldsymbol{p}) = J_j(\boldsymbol{p}) + \left[\boldsymbol{a}^T \backslash p + b\right]\left[\beta\boldsymbol{a}^T\boldsymbol{p} + c\right]$$

Proof. **part I**
The proof of this theorem mainly rely on the strict convexity of the dual problem, which impose that the optimum is unique for all parameters in feasible set. The continuity follows because any boundary of two regions belongs to both closure of the two regions. Since the optimum is unique, the solution across the boundary is continuous. By construction the

number of regions should be upper bounded by the number of all possible combinations of active constraints, which is finite and is worst case exponential in number of samples.

To see that critical regions constitute a partition, notice that since a feasible configuration of $\boldsymbol{p}$ admits a solution, it must be contained in one region, by existence of the solution. The region it belongs to must be unique. This can be seen by contradiction: Because in the interior of any two different regions, the set of active constraints are different, the optimum in two regions cannot be the same except for at the boundary. however, assume that a feasible configuration belongs to two regions, this means the dual problem has at least two optimum, which is contradictory to the uniqueness of the solution.

part II

Let $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ be two feasible parameters for the dual. Define

$$\begin{cases} \boldsymbol{p}_\beta = \beta \boldsymbol{p}_1 + (1-\beta)\boldsymbol{p}_2 \\ \boldsymbol{\alpha}^\beta = \beta \boldsymbol{\alpha}^*(\boldsymbol{p}_1) + (1-\beta)\boldsymbol{\alpha}^*(\boldsymbol{p}_2) \\ 0 \leq \beta \leq 1 \end{cases} \tag{4.62}$$

Note that the simplex constraints and the necessary and sufficient conditions on $\boldsymbol{p}$ imply that its feasible set is convex, hence $\boldsymbol{p}_\beta$ is also feasible. The feasibility of $\boldsymbol{\alpha}^\beta$ is also obvious. Now consider the chain of inequality:

$$\mathcal{J}(\boldsymbol{\alpha}^*(\beta \boldsymbol{p}_1 + (1-\beta)\boldsymbol{p}_2)) = \mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{p}_\beta)) \tag{4.63}$$

$$\leq \mathcal{J}(\boldsymbol{\alpha}^\beta) = \mathcal{J}(\beta \boldsymbol{\alpha}^*(\boldsymbol{p}_1) + (1-\beta)\boldsymbol{\alpha}^*(\boldsymbol{p}_2)) \tag{4.64}$$

$$\leq \beta \mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{p}_1)) + (1-\beta)\mathcal{J}(\boldsymbol{\alpha}^*(\boldsymbol{p}_2)) \tag{4.65}$$

The first inequality is because $\boldsymbol{\alpha}^*(\boldsymbol{p}_\beta)$ is the optimal (minimum) solution, and the second inequality is due to the convexity of the function $\mathcal{J}(\cdot)$. Hence the optimal value of the parameterized dual, as a function of parameter $\boldsymbol{p}$, is convex in the entire feasible set. In addition, there are only finite number of boundaries which has zero measure in the $\dim(\boldsymbol{p})$ space, hence the parameterized dual solution is almost everywhere differentiable. It's worth mentioning that the proof only relies on feasibility and the convexity of objective function, hence it can be generalized beyond the quadratic case.

part III

The proof relies on another geometric interpretation of the classic SVM. In fact the inner dual $\mathcal{J}$ can be rewritten as finding the polytope distance between the reduced convex hulls of the two classes of data-points, i.e., the dual is equivalent to

$$\begin{aligned} d(\boldsymbol{p}) = \min_{\boldsymbol{u},\boldsymbol{v}} \quad & \|\boldsymbol{u} - \boldsymbol{v}\|^2 \\ \text{s.t.} \quad & \begin{cases} \boldsymbol{u} \in \text{conv}_{\boldsymbol{p}}(\{\boldsymbol{u}_i \mid y_i = +1\}) \\ \boldsymbol{v} \in \text{conv}_{\boldsymbol{p}}(\{\boldsymbol{v}_i \mid y_i = -1\}) \end{cases} \end{aligned} \tag{4.66}$$

where for a finite point set $\mathcal{U} \in \mathbb{R}^d$, the reduced convex hall is

$$\text{conv}_{\boldsymbol{p}} = \left\{ \sum_{\boldsymbol{u} \in \mathcal{U}} \alpha_u u \;\mid\; 0 \leq \alpha_u \leq c_i p_i,\ \sum_{\boldsymbol{u} \in \mathcal{U}} \alpha_u = 1 \right\}.$$

We see that $d(\boldsymbol{p})$ may change its expression if and only if the optimal $\boldsymbol{u}^*$ or $\boldsymbol{u}^*$ changes from one face of the reduced convex hall to another face of different dimension. Let the neighboring expression of $\mathcal{J}$ be J_1 and J_2, which correspond to two faces of the reduced convex hall, say $\mathcal{F}_1$ and $\mathcal{F}_2$ respectively. Without loss of generality, assume that $\dim(\mathcal{F}_1) < \dim(\mathcal{F}_1)$, then $\mathcal{F}_1 \subset \mathcal{F}_1$, i.e., $\mathcal{F}_1$ is contained in the boundary of $\mathcal{F}_2$. Hence the distance expressions $d_1(\boldsymbol{p})$ and $d_1(\boldsymbol{p})$ must have

$$d_1(\boldsymbol{p}) - d_2(\boldsymbol{p}) \geq 0 \quad \forall\, \boldsymbol{p} \in \mathbb{R}^m \tag{4.67}$$

which is ony possible if $d_1(\boldsymbol{p}) - d_2(\boldsymbol{p})$ has positive semidefinite Hessian.

□

4.5.3 Lemma and Theorems for PDM: Global Optimization

Recall that we focus on the following non-smooth convex maximization:

$$\max_{p \in \mathbb{P}} \mathcal{F}(p) \tag{4.68}$$

Theorem. *$\boldsymbol{p}^*$ is a global optimal solution of the problem $\max_{\boldsymbol{p} \in \mathbb{P}} \mathcal{F}(\boldsymbol{p})$, if and only if for all $\boldsymbol{p} \in \mathbb{P}$, $\boldsymbol{q} \in E_{\mathcal{F}(\boldsymbol{p}^*)}$, $g(\boldsymbol{q}) \in \partial\mathcal{F}(\boldsymbol{q})$, we have*

$$(\boldsymbol{p} - \boldsymbol{q})^T g(\boldsymbol{q}) \leq 0 \tag{4.69}$$

where $\partial\mathcal{F}(\boldsymbol{q})$ is the set of subgradients of $\mathcal{F}$ at $\boldsymbol{p}$.

Proof. Necessity Assume $\boldsymbol{p}^*$ is a solution (global maximizer) of problem (4.68). Let $\boldsymbol{q}$ be a point in the level set $E_{\mathcal{F}(\boldsymbol{p}^*)}$, i.e., $\mathcal{F}(\boldsymbol{q}) = \mathcal{F}(\boldsymbol{p}^*)$. Then by the convexity of $\mathcal{F}$, we have

$$(\boldsymbol{p} - \boldsymbol{q})^T g(\boldsymbol{q}) \leq \mathcal{F}(\boldsymbol{p}) - \mathcal{F}(\boldsymbol{q}) = \mathcal{F}(\boldsymbol{p}) - \mathcal{F}(\boldsymbol{p}^*) \leq 0 \qquad \forall g(\boldsymbol{q}) \in \partial\mathcal{F}(\boldsymbol{q}) \tag{4.70}$$

where the first inequality is from the definition of sub-gradient.

Sufficiency

Proof by contradiction. Suppose $\boldsymbol{p}^*$ is not a solution (global maximizer) and it holds that

$$(\boldsymbol{p} - \boldsymbol{q})^T g(\boldsymbol{q}) \leq 0 \quad \forall \boldsymbol{q} \in E_{\mathcal{F}(\boldsymbol{p}^*)},\ \boldsymbol{p} \in \mathbb{P},\ g(\boldsymbol{q}) \in \partial\mathcal{F}(\boldsymbol{q}) \tag{4.71}$$

Then there exists some point $\boldsymbol{u} \in \mathbb{P}$ such that $\mathcal{F}(\boldsymbol{u}) > \mathcal{F}(\boldsymbol{p}^*)$. Now consider the epigraph

$$L_{\mathcal{F}(\boldsymbol{p}^*)} = \{\boldsymbol{p} \,:\, \mathcal{F}(\boldsymbol{p}) \leq \mathcal{F}(\boldsymbol{p}^*)\}$$

which is a closed convex set due to the convexity of $\mathcal{F}$. Denote the projection of $\boldsymbol{p}$ on $L_{\mathcal{F}(\boldsymbol{p}^*)}$ by $\boldsymbol{q}$, then we have

$$\|\boldsymbol{q} - \boldsymbol{u}\| = \min_{\boldsymbol{p} \in L_{\mathcal{F}(\boldsymbol{p}^*)}} \|\boldsymbol{p} - \boldsymbol{u}\| \tag{4.72}$$

Since $\boldsymbol{p}^*$ is not in $L_{\mathcal{F}(\boldsymbol{p}^*)}$, we have

$$\|\boldsymbol{q} - \boldsymbol{u}\| > 0 \tag{4.73}$$

strictly holds. Moreover, the global minimizer $\boldsymbol{p}_*$ cannot be this projection, i.e., $\boldsymbol{p}_* \neq \boldsymbol{q}$, otherwise $\min_{\boldsymbol{p}\in L} \|\boldsymbol{p}-\boldsymbol{u}\| = \|\boldsymbol{p}^*-\boldsymbol{u}\|$ for all epigraphs, which can only be true when $\mathcal{F}$ is a constant function.

In addition, the projection problem can be reformulated as the following least square form:

$$\begin{aligned} &\min\ d(\boldsymbol{p}) = \frac{1}{2}\|\boldsymbol{p}-\boldsymbol{u}\|^2 \\ &\text{s.t.}\quad \boldsymbol{p} \in L_{\mathcal{F}(\boldsymbol{p}^*)} \end{aligned} \tag{4.74}$$

By Slater's condition for nondifferentiable functions, $\boldsymbol{q}$ is a solution to problem (4.74) if and only if there exists a multiplier λ such that $(\lambda, \boldsymbol{q})$ is the solution to the following complementary problem:

$$\begin{cases} \lambda \geq 0 \\ 0 \in \nabla d(\boldsymbol{p}) + \lambda \partial \mathcal{F}(\boldsymbol{q}) \\ \lambda(\mathcal{F}(\boldsymbol{p}^*) - \mathcal{F}(\boldsymbol{q})) = 0 \\ \mathcal{F}(\boldsymbol{p}^*) - \mathcal{F}(\boldsymbol{q}) \geq 0 \end{cases} \tag{4.75}$$

Obviously $\lambda \neq 0$ otherwise $\nabla d(\boldsymbol{q}) = \boldsymbol{q}-\boldsymbol{u} = 0$ which contradicts (4.73). With $\lambda > 0$ we get that the Slater's condition is equivalent to

$$\begin{cases} \lambda > 0 \\ 0 \in \nabla d(\boldsymbol{q}) + \lambda \partial \mathcal{F}(\boldsymbol{q}) \\ \mathcal{F}(\boldsymbol{q}) = \mathcal{F}(\boldsymbol{p}^*) \end{cases} \tag{4.76}$$

since $\boldsymbol{p}_* \neq \boldsymbol{q}$ we know $0 \notin \partial\mathcal{F}(\boldsymbol{q})$, then the above conditions indicate that there exist some $g(\boldsymbol{q}) \in \partial\mathcal{F}(\boldsymbol{q})$ such that

$$\boldsymbol{q} - \boldsymbol{u} + \lambda g(\boldsymbol{q}) = 0 \tag{4.77}$$

hence we get $(\boldsymbol{u}-\boldsymbol{q})^T g(\boldsymbol{q}) = \frac{1}{\lambda}\|\boldsymbol{u}-\boldsymbol{q}\|^2 > 0$ which contradicts the assumption (4.71). □

Recall that we have obtained an approximate auxiliary problem as follows:

$$\max_{\boldsymbol{p}\in\mathbb{P},\ g(\boldsymbol{q}^i)\in\partial\mathcal{J}(\boldsymbol{q}^i)} (\boldsymbol{p}-\boldsymbol{q}^i)^T g(\boldsymbol{q}^i) \tag{4.78}$$

Proposition. *Problem (4.78) is equivalent to*

$$\max_{\boldsymbol{p}\in\mathbb{P}} \left\{ \max_{g(\boldsymbol{q}^i)\in V(\partial\mathcal{F}(\boldsymbol{q}^i))} (\boldsymbol{p}-\boldsymbol{q}^i)^T g(\boldsymbol{q}^i) \right\} \tag{4.79}$$

Proof. This is because for bilinear program having disjoint feasible sets, their solution must be on the vetices. See [Boyd]. □

Proposition. *For any $\boldsymbol{p} \in \mathbb{P}$, if there exist $\boldsymbol{q}^i \in A_{\boldsymbol{p}}^m$, $g(\boldsymbol{q}^i) \in V(\partial\mathcal{F}(\boldsymbol{q}^i))$, and $\boldsymbol{u}^i$ defined as*

$$(\boldsymbol{u}^i - \boldsymbol{q}^i)^T \boldsymbol{s}^i = \max_{\boldsymbol{p}\in\mathbb{P},\ g(\boldsymbol{q}^i)\in V(\partial\mathcal{F}(\boldsymbol{q}^i))} (\boldsymbol{p}-\boldsymbol{q}^i)^T g(\boldsymbol{q}^i) \tag{4.80}$$

such that $(\boldsymbol{u}^i-\boldsymbol{q}^i)^T g(\boldsymbol{q}^i) > 0$, then $\boldsymbol{u}^i$ improves the objective strictly, i.e., $\mathcal{F}(\boldsymbol{u}^i) > \mathcal{F}(\boldsymbol{p})$.

Proof. Since $\boldsymbol{u}^i$ is the solution to the auxiliary problem, we have

$$\max_{\boldsymbol{p}\in\mathbb{P}} \ (\boldsymbol{p}-\boldsymbol{q}^i)^T g(\boldsymbol{q}^i) = (\boldsymbol{u}^i-\boldsymbol{q}^i)^T g(\boldsymbol{q}^i) \tag{4.81}$$

and the lemma follows simply from the convexity:

$$\mathcal{F}(\boldsymbol{u}^i)-\mathcal{F}(\boldsymbol{p}) = \mathcal{F}(\boldsymbol{u}^i)-\mathcal{F}(\boldsymbol{q}^i) \geq (\boldsymbol{u}^i-\boldsymbol{q}^i)^T g(\boldsymbol{q}^i) > 0 \tag{4.82}$$

□

Lemma. *Let the global **minimizer** of $\mathcal{F}(\boldsymbol{p})$ be $\boldsymbol{p}_*$, then for $\boldsymbol{p}\neq\boldsymbol{p}_*$ and $\boldsymbol{h}\in\mathbb{R}^n$, there exist a **unique** positive scalar γ, such that $\boldsymbol{p}_*+\gamma\boldsymbol{h}\in E_{\mathcal{F}(\boldsymbol{p})}$.*

Proof. Existence: Proof by contradiction. Suppose that there is no number such that $\boldsymbol{p}_*+\gamma\boldsymbol{h}\in E_{\mathcal{J}(\boldsymbol{p})}$, then viewing $\mathcal{F}(\boldsymbol{p}_*+\gamma\boldsymbol{h})$ as " convex function restricted to a line", i.e., a 1D convex function of the variable γ, we have

$$\mathcal{F}(\boldsymbol{p}_*+\gamma\boldsymbol{h}) < \mathcal{F}(\boldsymbol{p}) \quad \forall\gamma\geq 0 \tag{4.83}$$

Consider $\text{epi}(\mathcal{F}) = \{(\boldsymbol{p},r) \ : \ \mathcal{F}(\boldsymbol{p})\leq r\}$, which is a convex set due to the convexity of $\mathcal{F}$. For $\gamma\geq 0$ it holds that $(\boldsymbol{p}_*+\gamma\boldsymbol{h},\mathcal{F}(\boldsymbol{p}))\in\text{epi}(\mathcal{F})$ due to (4.83).

Now we show that $(\boldsymbol{h},0)$ is a direction of $\text{epi}(\mathcal{F})$. Again by contradiction suppose this is not true and there exists a point $\boldsymbol{y}\in\text{epi}(\mathcal{F})$ and a positive number β such that $\boldsymbol{y}+\beta(\boldsymbol{h},0)\in\mathbb{R}^{m+1}\setminus\text{epi}(\mathcal{F})$. Because $\mathbb{R}^{m+1}\setminus\text{epi}(\mathcal{F})$ is an open set, we can find a scalar μ that satisfies the following condition:

$$\mu(\boldsymbol{p}_*,\mathcal{F}(\boldsymbol{p}))+(1-\mu)(\boldsymbol{y}+\beta(\boldsymbol{h},0))\in\mathbb{R}^{m+1}\setminus\text{epi}(\mathcal{F}), \quad 0<\mu<1 \tag{4.84}$$

However it must also be the case that $\mu(\boldsymbol{p}_*,\mathcal{F}(\boldsymbol{p}))+(1-\mu)(\boldsymbol{y}+\beta(\boldsymbol{h},0))$ lies on the line segment joining some two points of $\text{epi}(\mathcal{F})$. Consider points $(\boldsymbol{p}_*,\mathcal{F}(\boldsymbol{p}))+\frac{(1-\mu)\beta}{\mu}(\boldsymbol{h},0)$ and $\boldsymbol{y}$, the following holds:

$$\mu\left((\boldsymbol{p}_*,\mathcal{F}(\boldsymbol{p}))+\frac{(1-\mu)\beta}{\mu}(\boldsymbol{h},0)\right)+(1-\mu)\boldsymbol{y} = \mu(\boldsymbol{p}_*,\mathcal{F}(\boldsymbol{p}))+(1-\mu)(\boldsymbol{y}+\beta(\boldsymbol{h},0)) \tag{4.85}$$

which belongs to $\text{epi}(\mathcal{F})$ again by the convexity of $\text{epi}(\mathcal{F})$, but is contradictory to (4.84). Hence $(\boldsymbol{h},0)$ must be a direction of $\text{epi}(\mathcal{F})$. Consider moving point $(\boldsymbol{p}_*,\mathcal{F}(\boldsymbol{p}))$ with this direction, we get

$$(\boldsymbol{p}_*,\mathcal{F}(\boldsymbol{p}))+\gamma(\boldsymbol{h},0)\in\text{epi}(\mathcal{F}) \quad \forall\gamma\geq 0 \tag{4.86}$$

which implies that $\boldsymbol{p}_*+\gamma\boldsymbol{h}$ is also the global minimum of $\mathcal{F}$ for any $\gamma\geq 0$, contradicting the strict convexity of $\mathcal{F}$.

Uniqueness

Assume two possible γ_1 and γ_2 exist, such that $\boldsymbol{p}_* + \gamma_i \boldsymbol{h} \in E_{\mathcal{F}(\boldsymbol{p})}$. WLOG assume further $0 < \gamma_1 \leq \gamma_2$, with the convexity of $\mathcal{F}$ we have

$$\mathcal{F}(\boldsymbol{p}) = \mathcal{F}(\boldsymbol{p}_* + \gamma_1 \boldsymbol{h}) = \mathcal{F}\left((1 - \frac{\gamma_1}{\gamma_2})\boldsymbol{p}_* + \frac{\gamma_1}{\gamma_2}(\boldsymbol{p}_* + \gamma_2 \boldsymbol{h})\right) \tag{4.87}$$

$$\leq \left(1 - \frac{\gamma_1}{\gamma_2}\right)\mathcal{F}(\boldsymbol{p}_*) + \frac{\gamma_1}{\gamma_2}\mathcal{F}(\boldsymbol{p}_* + \gamma_2 \boldsymbol{h}) \tag{4.88}$$

$$= \left(1 - \frac{\gamma_1}{\gamma_2}\right)\mathcal{F}(\boldsymbol{p}_*) + \frac{\gamma_1}{\gamma_2}\mathcal{F}(\boldsymbol{p}) \tag{4.89}$$

$$\leq \mathcal{F}(\boldsymbol{p}) \tag{4.90}$$

which is only possible when all equality holds, i.e., when $\gamma_1 = \gamma_2$. □

Theorem. *Algorithm 4 generates a sequence* $\{\boldsymbol{p}^{(1)}, \cdots, \boldsymbol{p}^{(k)}, \cdots\}$ *having non-decreasing function values. The sequence converges to an approximate maximizer of* $\mathcal{F}(\boldsymbol{p})$ *in a finite number of steps.*

Proof. We first show that the condition on $\Delta A^m_{\boldsymbol{p}} \triangleq= (\boldsymbol{u}^{(k)} - \boldsymbol{q}^{i^*})^T \boldsymbol{s}^{i^*}$ guarantees the improvement of the objective function. In fact, since

$$i^* = \text{argmax}_{i=1,\cdots,m}\{(\boldsymbol{u}^i - \boldsymbol{q}^i)^T \boldsymbol{s}^i\} \tag{4.91}$$

and $\boldsymbol{u}^{(k)} = \boldsymbol{u}^{i^*}$, we have

$$\Delta A^m_{\boldsymbol{p}} = (\boldsymbol{u}^{(k)} - \boldsymbol{q}^{i^*})^T \boldsymbol{s}^{i^*} > 0 \tag{4.92}$$

Because $\boldsymbol{q}^{i^*} \in A^m_{\boldsymbol{r}^{(k)}}$ and $\boldsymbol{s}^{i^*} \in V(\partial\mathcal{F}(\boldsymbol{q}^{i^*}))$, using Proposition 4.5.3 we get

$$\mathcal{F}(\boldsymbol{u}^{(k)}) > \mathcal{F}(\boldsymbol{r}^{(k)}) \tag{4.93}$$

and according to the algorithm, we further have

$$\mathcal{F}(\boldsymbol{p}^{(k+1)}) = \mathcal{F}(\boldsymbol{u}^{(k)}) > \mathcal{F}(\boldsymbol{r}^{(k)}) \geq \mathcal{F}(\boldsymbol{p}^{(k)}) \tag{4.94}$$

The last inequality is because a local maximizer is used to find a local solution $\boldsymbol{r}^{(k)}$ starting from $\boldsymbol{p}^{(k)}$. Since the number of local solutions (vertices of $\mathbb{P}$) are finite, this sequence reaches the global maximizer in a finite number of steps, or stops at an approximate solution where improvement could not be found at current approximation degree. It is worth mentioning that the above argument holds even if the approximate level set are obtained numerically by finding the root of the following function:

$$\Phi(\gamma_i) \triangleq \mathcal{F}(\boldsymbol{p}_* + \gamma_i \boldsymbol{h}^i) - \mathcal{F}(\boldsymbol{r}^{(k)}) = 0 \tag{4.95}$$

we can simply choose γ_i, such that the approximate root satisfies

$$\mathcal{F}(\boldsymbol{q}^i) = \mathcal{F}(\boldsymbol{p}_* + \gamma_i \boldsymbol{h}^i) \geq \mathcal{F}(\boldsymbol{r}^{(k)}) \tag{4.96}$$

Then we get

$$\mathcal{F}(\boldsymbol{u}^{(k)}) - \mathcal{F}(\boldsymbol{r}^{(k)}) \geq \mathcal{F}(\boldsymbol{u}^{(k)}) - \mathcal{F}(\boldsymbol{q}^{i^*}) \geq (\boldsymbol{u}^{(k)} - \boldsymbol{q}^{i^*})^T \boldsymbol{s}^{i^*} > 0 \tag{4.97}$$

which is the same as (4.93). □

4.5.3.1 Finding the Global Minimum with Sub-gradient Descent

PDM requires the knowledge of the global minimizer of the non-smooth convex function $\mathcal{F}$. In this section, we show that a simple sub-gradient descent algorithm can be used to efficiently find the global minimum.

Theorem. *Sub-Gradient Descent for Finding Global Minimum*
Let $\sup_{\boldsymbol{p}} ||\boldsymbol{p}^{(1)} - \boldsymbol{p}|| = B$, *and the Lipschitz constant of* $\mathcal{F}(\boldsymbol{\alpha}^*(\boldsymbol{p}))$ *be* G, *then sub-gradient descent with iteration* T *and optimal step size* $\tau_i = B/G\sqrt{T}\ \ \forall i$ *converges to global minimum within* $O\left(1/\sqrt{T}\right)$. *To be specific, let* $\mathcal{F}_*$ *be the global minimum of the learning objective (IO), then*

$$\mathcal{F}(\boldsymbol{\alpha}^*(\boldsymbol{p}_{best}^{(T)})) - \mathcal{F}_* \leq \frac{BG}{\sqrt{T}} \leq \frac{4C^3|\lambda_{\min}(H)|}{\lambda_{\min}(Q)|\lambda_{\max}(R)|}\frac{N\sqrt{d}}{\sqrt{T}}, \qquad \textit{where} \tag{4.98}$$

$\mathcal{F}(\boldsymbol{\alpha}^*(\boldsymbol{p}_{best}^{(T)})) \triangleq \min\left\{\mathcal{F}(\boldsymbol{\alpha}^*(\boldsymbol{p}^{(1)})), \cdots, \mathcal{F}(\boldsymbol{\alpha}^*(\boldsymbol{p}^{(T)}))\right\}$

Proof. The proof of convergence resembles that for classic sub-gradient method. In short, we note that

$$||\boldsymbol{p}^{(i+1)} - \boldsymbol{p}^*||_2^2 = ||\text{Proj}(\boldsymbol{p}^{(i)} - \tau_i g^{(i)}) - \boldsymbol{p}^*||_2^2 \tag{4.99}$$

$$\leq ||\boldsymbol{p}^{(i)} - \boldsymbol{p}^*||_2^2 + \tau_i^2||g^{(i)}||_2^2 - 2\tau_i\left[\mathcal{F}(\boldsymbol{\alpha}^*(\boldsymbol{p}^{(i)})) - \mathcal{F}(\boldsymbol{\alpha}^*(\boldsymbol{p}^*))\right] \tag{4.100}$$

by applying the definition of sub-gradient to the convex function $\mathcal{F}(\boldsymbol{\alpha}^*(\boldsymbol{p}))$. Telescoping the above inequality,

$$||\boldsymbol{p}^{(T+1)} - \boldsymbol{p}^*||_2^2 \tag{4.101}$$

$$\leq ||\boldsymbol{p}^{(1)} - \boldsymbol{p}^*||_2^2 + \sum_{i=1}^{T}\tau_i^2||g^{(i)}||_2^2 - 2\sum_{i=1}^{T}\tau_i\left[\mathcal{F}(\boldsymbol{\alpha}^*(\boldsymbol{p}^{(i)})) - \mathcal{F}(\boldsymbol{\alpha}^*(\boldsymbol{p}^*))\right] \tag{4.102}$$

Rearranging terms and using the fact $||\boldsymbol{p}^{(1)} - \boldsymbol{p}^*||_2^2 \leq B^2$, $||g^{(i)}||_2^2 \leq G^2$, $||\boldsymbol{p}^{(T+1)} - \boldsymbol{p}^*||_2^2 \geq 0$ we get

$$2\sum_{i=1}^{T}\tau_i\left[\mathcal{F}(\boldsymbol{\alpha}^*(\boldsymbol{p}^{(i)})) - \mathcal{F}(\boldsymbol{\alpha}^*(\boldsymbol{p}_*))\right] \leq B^2 + G^2\sum_{i=1}^{T}\tau_i^2 \tag{4.103}$$

By the definition of $\mathcal{F}(\boldsymbol{\alpha}^*(\boldsymbol{p}_{best}^{(T)}))$ and $\mathcal{F}_*$,

$$\mathcal{F}(\boldsymbol{\alpha}^*(\boldsymbol{p}_{best}^{(T)2})) - \mathcal{F}_* \leq \frac{B^2 + G^2\sum_{i=1}^{T}\tau_i^2}{2\sum_{i=1}^{T}\tau_i} \tag{4.104}$$

Now we show the the constant step size $\tau_i = \frac{B}{G\sqrt{T}}$ is optimal, in the sense that it minimizes the above convergence upper bound. Consider the the following chain of inequalities.

$$\frac{B^2 + G^2 \sum_{i=1}^T \tau_i^2}{2\sum_{i=1}^T \tau_i} \geq \frac{B^2 + G^2 \sum_{i=1}^T \tau_i^2}{2\sqrt{T\sum_{i=1}^n \tau_i^2}} \tag{4.105}$$

$$= \frac{1}{2\sqrt{T}} \left(\frac{B^2}{\sqrt{\sum_{i=1}^T \tau_i^2}} + G^2 \sqrt{\sum_{i=1}^T \tau_i^2} \right) \geq \frac{BG}{\sqrt{T}} \tag{4.106}$$

$$\tag{4.107}$$

the first inequality is due to Cauchy-Schwarz, with equality when τ_is and 1s are co-linear. The last inequality is equal when $\sum_{i=1}^T \tau_i^2 = B^2/G^2$. Combining the two equality requirement, we get that the step size

$$\tau_i = \frac{B}{G\sqrt{T}} \qquad \forall i = 1, \cdots, T$$

minimizes the upper bound. In order to further bound the sub-gradient performance in terms of problem size, note that since $0 \leq \boldsymbol{p}_i \leq 1$, we have $B \leq \sqrt{d}$, where d is the dimension of $\boldsymbol{p}$. On the other hand, we have

$$\frac{\partial}{\partial \boldsymbol{p}} \frac{1}{2} \boldsymbol{\alpha}^T \boldsymbol{Q} \boldsymbol{\alpha} = (\boldsymbol{C}_{\mathcal{A}}^p)^T \boldsymbol{R}^{-1} \boldsymbol{C}_{\mathcal{A}}^{\alpha} \boldsymbol{H} \boldsymbol{Q}^{-1} \boldsymbol{\alpha} \tag{4.108}$$

Thus with operation norm inequality, we have

$$G \leq \|\boldsymbol{C}_{\mathcal{A}}^p\| \cdot \|\backslash R^{-1}\| \cdot \|\boldsymbol{C}_{\mathcal{A}}^{\alpha}\| \cdot \|\boldsymbol{H}\| \cdot \|\boldsymbol{Q}^{-1}\| \cdot \|\boldsymbol{\alpha}\| \tag{4.109}$$

$$\leq 2C^2 \cdot \frac{1}{|\lambda_{\max}(R)|} \cdot 2 \cdot |\lambda_{\min}(H)| \cdot \frac{1}{\lambda_{\min}(Q)} \cdot C\sqrt{N} \tag{4.110}$$

$$= \frac{4C^3 |\lambda_{\min}(H)|}{\lambda_{\min}(Q)|\lambda_{\max}(R)|} \sqrt{N} \tag{4.111}$$

where $C \triangleq \max\{c_i\}$ and we have used $||A|| = \sqrt{\lambda_{\max}(A^T A)}$ and the following fact that: $(\boldsymbol{C}_{\mathcal{A}}^p)^T \boldsymbol{C}_{\mathcal{A}}^p = 2\text{diag}\{c_1^2, \cdots, c_m^2\}$; $(\boldsymbol{C}_{\mathcal{A}}^{\alpha})^T \boldsymbol{C}_{\mathcal{A}}^{\alpha} = 2\backslash I$; $\boldsymbol{R}$ and $\boldsymbol{H}$ are negative symmetric definite and $0 \leq \alpha_i \leq c_i$. □

4.5.3.2 More Reformulation Examples

In this section, we provide more examples on how large margin learning variations could be reformulated into OPT1 in the main test. More importantly, we give detailed proofs for the parametric analysis of the dual problem.

Example 2. *Consider the learning objective of Semi Supervised Support Vector Machine (S3VM):*

$$\min_{\boldsymbol{w}, b, \widehat{\boldsymbol{y}}_u} \frac{1}{2} ||\boldsymbol{w}||_{\mathcal{H}}^2 + C_1 \sum_{i=1}^{l} V(y_i, h_i) + C_2 \sum_{i=l+1}^{n} V(\widehat{y}_i, h_i) \tag{4.112}$$

where l is the number of labeled samples and $n-l$ unlabeled samples are included in the loss with "tentative" label $\widehat{\boldsymbol{y}}_u$, which constitute additional variables to minimize over. Note the following interesting equivalent form:

$$\min_{\boldsymbol{w},b}\min_{\boldsymbol{p}}\ \frac{1}{2}||\boldsymbol{w}||_{\mathcal{H}}^2 + C_1\sum_{i=1}^{l}V(y_i,h_i) + C_2\sum_{i=l+1}^{n}p_iV(1,h_i) + (1-p_i)V(-1,h_i) \tag{4.113}$$

The equivalence is due to the fact that minimizing over p_i will cause all its mass to concentrate on the smaller of $V(1,h_i)$ and $V(-1,h_i)$. Formally for any variables $\xi_1,\cdots,\xi_M$ we have

$$\min_m\{\xi_1,\cdots,\xi_M\} = \min_{\boldsymbol{p}\in\mathbb{S}^M}\sum_{m=1}^{M}p_m\xi_m,$$

where $\mathbb{S}^M$ is the simplex in $\mathbb{R}^M$. Since (4.113) is strictly feasible and biconvex in $(\boldsymbol{w},b)$ and $\boldsymbol{p}$, we can safely exchange the order of minimization and obtain (OPT1) as an equivalent form to (4.112). The newly introduce variable p_i can be interpreted as the "probability" of $\widehat{y}_i = 1$.

Example. *Consider VCMKL*

$$\begin{aligned}\min_{\boldsymbol{w}_m,b_m}\quad & \frac{1}{2}\sum_{m=1}^{M}||\boldsymbol{w}_m||^2 \\ & + C_1\sum_{i\in I^+}\max_m\{[1-y_i(\boldsymbol{w}_m\cdot\boldsymbol{x}_i+b_m)]_+\} \\ & + C_2\sum_{i\in I^-}\min_m\{[1-y_i(\boldsymbol{w}_m\cdot\boldsymbol{x}_i+b_m)]_+\}\end{aligned} \tag{4.114}$$

Introducing hidden state variables $\boldsymbol{p}$ with the same trick, we get $\min_m\{\xi_1,\cdots,\xi_M\} = \min_{\boldsymbol{p}\in\mathbb{S}^M}\ \sum_{m=1}^{M}p_m\xi_m$, for the last term and with the same argument to justify the exchange of minimization orders, the original learning problem is equivalent to

$$\begin{aligned}\min_{\boldsymbol{p}_i\in\mathbb{S}^M}\min_{\boldsymbol{w}_m,b_m}\quad & \frac{1}{2}\sum_{m=1}^{M}||\boldsymbol{w}_m||^2 \\ & + C_1\sum_{i\in I^+}\max_m\{[1-y_i(\boldsymbol{w}_m\cdot\boldsymbol{x}_i+b_m)]_+\} \\ & + C_2\sum_{i\in I^-}\sum_{m=1}^{M}p_{im}[1-y_i(\boldsymbol{w}_m\cdot\boldsymbol{x}_i+b_m)]_+\end{aligned} \tag{4.115}$$

As the context implies, the newly introduced variables can be thought of as indicators for hidden states. Obviously, (4.115) has the same form as OPT1.

Example. *As a final example, consider robust SVM proposed in xu2006robust with ramp loss*

$$\min_{\boldsymbol{w}} \frac{1}{2}||\boldsymbol{w}||^2 + \sum_i \min\{1, [1 - y_i \boldsymbol{w} \cdot \boldsymbol{x_i}]_+\} \tag{4.116}$$

The non-convex robust loss essentially truncates Hinge loss by 1. Same argument applies and the training objective can be rewritten as

$$\min_{\boldsymbol{p}_i \in \mathbb{S}^2} \min_{\boldsymbol{w}} \frac{1}{2}||\boldsymbol{w}||^2 + \sum_i \{p_i[1 - y_i \boldsymbol{w} \cdot \boldsymbol{x}_i]_+ + 1 - p_i\} \tag{4.117}$$

We see that the loss terms will not exceed 1, which provide robustness against training outliers.

Recall that the dual of the inner minimization has the form

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i y_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) y_j \alpha_j \\ \text{subject to} \quad & 0 \le \alpha_i \le c_i p_i \quad \text{for} \quad i = 1, \dots, n \\ & \boldsymbol{\alpha}^T \boldsymbol{y} = 0 \end{aligned} \tag{4.118}$$

which can be encapsulated into a matrix form and with the common convex programming convention we consider the following equivalent problem:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \mathcal{J}(\boldsymbol{\alpha}; \boldsymbol{p}) = \frac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{Q} \boldsymbol{\alpha} - \mathbf{1}^T \boldsymbol{\alpha} \\ & \text{subject to} \begin{cases} \boldsymbol{C}^\alpha \boldsymbol{\alpha} \le \boldsymbol{C}^p \boldsymbol{p} + \boldsymbol{C}^0 \\ \boldsymbol{\alpha}^T \boldsymbol{y} = 0, \end{cases} \end{aligned} \tag{Dual}$$

For example for S^{3}VM, one has

$$\begin{aligned} \boldsymbol{x} &\triangleq [x_1; \cdots; x_n, \underbrace{x_1; \cdots; x_n}_{m}, \underbrace{x_1; \cdots; x_n}_{m}] \\ \boldsymbol{y} &\triangleq [y_1, \cdots, y_n, \underbrace{1, \cdots, 1}_{m}, \underbrace{-1, \cdots, -1}_{m}]^T \end{aligned}$$

Then $Q_{i,j} = \boldsymbol{y}_i \backslash y_j \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and the matrices $\boldsymbol{C}^\alpha, \boldsymbol{C}^p, \boldsymbol{C}^0$ in the constraints can be specified as follows with blocks

$$\boldsymbol{C}^\alpha = \begin{bmatrix} -\boldsymbol{I}_{(n+2m)\times(n+2m)} \\ \boldsymbol{I}_{(n+2m)\times(n+2m)} \end{bmatrix} \quad \boldsymbol{C}^p = \begin{bmatrix} \mathbf{0}_{(n+2m)\times m} \\ \mathbf{0}_{n\times m} \\ c_2 \boldsymbol{I}_{m\times m} \\ -c_2 \boldsymbol{I}_{m\times m} \end{bmatrix} \quad \boldsymbol{C}^0 = \begin{bmatrix} \mathbf{0}_{(n+2m)\times 1} \\ c_1 \mathbf{1}_{n\times 1} \\ \mathbf{0}_{m\times 1} \\ c_2 \mathbf{1}_{m\times 1} \end{bmatrix} \tag{4.119}$$

4.5.3.3 A decomposition Technique for non-Strictly Positive Definite Problems

We use a decomposition and null space method to deal with the case when $\boldsymbol{Q}$ is only positive symmetric semi-definite. Let the eigen-decomposition of $\boldsymbol{Q}$ be $\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\boldsymbol{T}}$, where columns of $\boldsymbol{U}$ are orthonormal basis of $\mathbb{R}^n$. The diagonal matrix $\Lambda = \begin{bmatrix} \boldsymbol{W} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$ in which $\boldsymbol{W}$ is a $k \times k$ diagonal matrix containing k non-zeros eigenvalues of $\boldsymbol{Q}$. Let $\boldsymbol{U} = [\boldsymbol{U}_1, \boldsymbol{U}_2]$ with columns of $\boldsymbol{U}_1$ containing the k eigenvectors associated with non-zeros eigenvalues. Consider the transformation $\chi_1 \triangleq \boldsymbol{U}_1^T\boldsymbol{\alpha}$ and $\chi_2 \triangleq \boldsymbol{U}_2^T\boldsymbol{\alpha}$, then problem is equivalent to

$$\begin{aligned} \min_{\chi_1,\chi_2} \quad & \mathcal{J}(\chi) = \frac{1}{2}\chi_1^T\boldsymbol{W}\chi_1 + \boldsymbol{r}_1^T\chi_1 + \boldsymbol{r}_2^T\chi_2 \\ \text{subject to} \quad & \begin{cases} \boldsymbol{C}^\alpha\boldsymbol{U}_1\chi_1 + \boldsymbol{C}^\alpha\boldsymbol{U}_2\chi_2 \leq \boldsymbol{C}^p\boldsymbol{p} + \boldsymbol{C}^0 \\ \boldsymbol{y}^T\boldsymbol{U}_1\chi_1 + \boldsymbol{y}^T\boldsymbol{U}_2\chi_2 = 0 \end{cases} \end{aligned} \tag{4.120}$$

where $\boldsymbol{r}_1 = -\boldsymbol{U}_1^T\boldsymbol{r}$ and $\boldsymbol{r}_2 = -\boldsymbol{U}_2^T\boldsymbol{r}$. Let us also define $D_1 = \begin{bmatrix} \boldsymbol{C}^\alpha\boldsymbol{U}_1 \\ \boldsymbol{y}^T\boldsymbol{U}_1 \end{bmatrix}$, $D_2 = \begin{bmatrix} \boldsymbol{C}^\alpha\boldsymbol{U}_2 \\ \boldsymbol{y}^T\boldsymbol{U}_2 \end{bmatrix}$, $D^p = \begin{bmatrix} \boldsymbol{C}^p \\ \boldsymbol{0} \end{bmatrix}$, $D^0 = \begin{bmatrix} \boldsymbol{C}^0 \\ \boldsymbol{0} \end{bmatrix}$ and $\boldsymbol{\eta} = [\boldsymbol{\mu}, \zeta]^T$, we get from the first KKT condition:

$$\boldsymbol{W}\chi_1 + \boldsymbol{r}_1 + D_1\boldsymbol{\eta} = 0 \tag{4.121a}$$

$$\boldsymbol{r}_2 + D_2\boldsymbol{\eta} = 0 \tag{4.121b}$$

The first equation yields $\chi_1 = -\boldsymbol{W}^{-1}(D_1^T\boldsymbol{\eta} + \boldsymbol{r}_1)$. With a similar active set argument as in theorem 1, we obtain

$$\chi_1 = -\boldsymbol{W}^{-1}(D_{1\mathcal{A}}^T\boldsymbol{\eta}_\mathcal{A} + \boldsymbol{r}_1) \tag{4.122}$$

$$D_{1\mathcal{A}}\chi_1 + D_{2\mathcal{A}}\chi_2 = D_\mathcal{A}^p\boldsymbol{p} + D_\mathcal{A}^0 \tag{4.123}$$

Now we adopt a null space method to solve $\boldsymbol{\eta}_\mathcal{A}$. Let the null space of $D_{2\mathcal{A}}^T$ be spanned by $n - m$ column vectors contained in matrix $Z_\mathcal{A}$, and let $Y_\mathcal{A}$ be any $n \times m$ matrix such that $[Y_\mathcal{A}, Z_\mathcal{A}]$ is full rank. Then we must have that the $(n-m) \times (n-m)$ matrix $D_{2\mathcal{A}}^T Y_\mathcal{A}$ is full rank and invertible, and $D_{2\mathcal{A}}^T Z_\mathcal{A} = \boldsymbol{0}$. Moreover we can write

$$\boldsymbol{\eta}_\mathcal{A} = Y_\mathcal{A}\boldsymbol{\eta}_Y + Z_\mathcal{A}\boldsymbol{\eta}_Z \tag{4.124}$$

as the sum of two components. With that equation (4.121b) becomes $D_{2\mathcal{A}}^T Y_\mathcal{A}\boldsymbol{\eta}_Y + \boldsymbol{r}_2 = 0$, thus

$$\boldsymbol{\eta}_Y = -(D_{2\mathcal{A}}^T Y_\mathcal{A})^{-1}\boldsymbol{r}_2 \tag{4.125}$$

Replace χ_1 in (4.123) by using (4.122)(4.124)(4.125), and multiple both side of (4.123) with $Z_\mathcal{A}^T$ (Recall $Z_\mathcal{A}^T D_{2\mathcal{A}} = \boldsymbol{0}$), we get

$$\begin{aligned} -\left(Z_\mathcal{A}^T D_{1\mathcal{A}}\boldsymbol{W}^{-1}D_{1\mathcal{A}}^T Z_\mathcal{A}\right)\boldsymbol{\eta}_Z = {} & Z_\mathcal{A}^T D_\mathcal{A}^p\boldsymbol{p} + Z_\mathcal{A}^T D_{1\mathcal{A}}\boldsymbol{W}^{-1}\boldsymbol{r}_1 \\ & - Z_\mathcal{A}^T D_{1\mathcal{A}}\boldsymbol{W}^{-1}D_{1\mathcal{A}}^T Y_\mathcal{A}(D_{2\mathcal{A}}^T Y_\mathcal{A})^{-1}\boldsymbol{r}_2 + Z_\mathcal{A}^T D_{1\mathcal{A}}^0 \end{aligned} \tag{4.126}$$

Define $\boldsymbol{G} = Z_{\mathcal{A}}^T D_{1\mathcal{A}} \boldsymbol{W}^{-1} D_{1\mathcal{A}}^T Z_{\mathcal{A}}$, whose invertibility under non-trivial classification can be showed similarly following the proof of lemma 1. Denote the last 3 terms in (4.126) as $\boldsymbol{\rho}_Z$,

$$\boldsymbol{\eta}_Z = -\boldsymbol{G}^{-1}(Z_{\mathcal{A}}^T D_{\mathcal{A}}^p \boldsymbol{p} + \boldsymbol{\rho}_Z) \tag{4.127}$$

To get χ_2, multiply both sides of (4.123) with $Y_{\mathcal{A}}^T$ and use results obtained so far,

$$\chi_2 = (Y_{\mathcal{A}}^T D_{2\mathcal{A}})^{-1} \left(D_{\mathcal{A}}^p \boldsymbol{p} - Y_{\mathcal{A}}^T D_{1\mathcal{A}} \chi_1 - D_{\mathcal{A}}^0 \right) \tag{4.128}$$

The critical region is characterized by

$$\begin{aligned} &\boldsymbol{\mu}_{\mathcal{A}} \geq 0 \\ &D_{1\mathcal{A}^c}\chi_1 + D_{2\mathcal{A}^c}\chi_2 \leq D_{\mathcal{A}^c}^p \boldsymbol{p} + D_{\mathcal{A}^c}^0 \end{aligned} \tag{4.129}$$

Noticeably, the optimal solution and the Lagrangian multipliers are no longer unique but depend on the choice of subspace basis $Y_{\mathcal{A}}$ and $Z_{\mathcal{A}}$. A simple fix is to adopt the projection method to obtain the "best" parametric solution and multipliers.

4.5.3.4 Critical Region Approximation

The most computational expensive step is the inner QP solver. By theorem 1, if $\boldsymbol{p}$ is in the critical regions that have been explored before, all information could be retrieved in an explicit form and there is no need to solve the inner problem again. However, when the variable goes to a new critical region, a QP solver has to be invoked for optimal solution and corresponding constraint partition.

Although it has been shown that even in the one dimensional case the number of critical regions is worst case exponential to the sample size of the problem, one can develop approximate parametric solutions that produces fewer (hence larger) critical regions. To further accelerate the algorithm by reducing the number of calls of the quadratic solver, we adopt the idea proposed in [186] that relaxes the original sample partition conditions to obtain "larger" critical regions. The relaxed sample partition condition is defined as

$$\begin{aligned} &y_i f_i \geq 1 - \epsilon_1,\ \alpha_i \in [-\epsilon_2, 0] \ \Rightarrow\ i \in \mathcal{O} \\ &y_i f_i \in [1 - \epsilon_1, 1 + \epsilon_1],\ \alpha_i \in [-\epsilon_2, c_i p_i + \epsilon_2] \ \Rightarrow\ i \in \mathcal{S}_u \\ &y_i f_i \leq 1 + \epsilon_1,\ \alpha_i \in [c_i p_i, c_i p_i + \epsilon_2] \ \Rightarrow\ i \in \mathcal{S}_b \end{aligned} \tag{4.130}$$

Theorem. *Inspired by [186]*
The approximate solution defined in (4.130) is the optimal solution of the following perturbed problem

$$\max_{\boldsymbol{\alpha}} \quad (\mathbf{1} + \boldsymbol{\eta})^T \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^T \boldsymbol{Q} \boldsymbol{\alpha}$$
$$\text{subject to} \begin{cases} \boldsymbol{C}^{\alpha} \boldsymbol{\alpha} \leq \boldsymbol{C}^{\theta} \boldsymbol{\theta} + \boldsymbol{C}^0 + \widetilde{\boldsymbol{C}}^0 \\ \boldsymbol{\alpha}^T \boldsymbol{y} = 0, \end{cases} \tag{4.131}$$

where $-\epsilon_1 \mathbf{1} \leq \boldsymbol{\eta} \leq \epsilon_1 \mathbf{1}$ *and* $0 \leq \widetilde{\boldsymbol{C}}^0 \leq \epsilon_2 \mathbf{1}$

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In conclusion, the second chapter started with two formulations of directed information based subset selection problem, and then submodularity analysis is provided for both objective functions. Seeing that the first objective is not monotonic, we adopted a randomized version of greedy heuristic. Since the second objective lacks submodularity, we introduced an novel notion, namely submodularity index, to characterize the "degree" of submodularity for general set functions. More importantly, we show that with SmI, the theoretical performance guarantee of the greedy heuristic can be naturally extended to a much broader class of functions. We also point out the connection between causal subset selection and the structure learning of causal graphs, based on which an efficient causal structure learning algorithm is established. Experimental results on synthesis and real data sets reaffirmed our theoretical findings, and also demonstrated the effectiveness of the proposed method for building the structure of causal graphs.

Provided with the correlation structure, the next chapter is focused on the learning outlier and novelty from multiple time series. The key idea is to incorporate both temporal dependence and inter-series relatedness for the construction of two "smoothing filters". The first non-parametric method can be viewed as a multitask version [207] of the classical non-parametric regression method. It is shown in this chapter that the learning formulation can be extended to handle data with exponential family distribution, and an efficient RBCD algorithm can be use to solve the convex optimization problem. The second method, CHMM, is inspired by collaborative filtering and linear system optimal filtering. Essentially, CHMM can be viewed as either "temporal constrained matrix factorization", or "Kalman filter incorporating inter-series correlation". The learning of CHMM is resolved with an EM algorithm by exploiting the structure of the graphical model.

The problem of learning system requirement for agile operation and optimal control is considered in the last chapter. Motivated by the needs to model system operation constraints with "optimization friendly" functions, we start by establishing a piece-wise convex classifier

and the corresponding learning formulation. Then we extend the classifier to a more general veto-consensus multiple kernel learning framework for fault detection, domain description, and semi-supervised event diagnosis. The main contribution of this chapter is the global optimization procedure parametric dual maximization. Both theoretical analysis and experimental results show that PDM outperforms other alternative optimization methods in solving a class of modified machine learning problems having non-convex objectives. Moreover, we provide two case studies that demonstrate the usage of the proposed ML schemes for CPS applications.

5.2 Future Work

The theoretical analysis and algorithms discussed for causal subset selection can be naturally extended to an online setting by adopting the notion of adaptive submodularity [208, 209, 210]. This will give rise to algorithms for online variable selection, sensor placement, and graph structure learning. Yet another direction for future work is to study further the concept of approximate submodularity. Very recently, the authors of [211] have summarized different approaches proposed so far and suggested several future directions. In additional to their proposal, we believe that the study of approximate submodularity for higher order greedy algorithms deserves more research effort: Since higher order greedy can be viewed as better level of approximation, an submodularity index in this context can help us understand more about "when/why greedy heuristics works".

The proposed RBCD algorithm bears some interesting features. It resembles SGD in that the expectation of each update is equal to the gradient, while unlike SGD the RBCD ensures a decrease of the objective function in each iteration. Arguably, recent machine learning literature focuses more on gradient descent based method but block coordinate descent seems to be ignored. This work advocates the use of BCD for a broader class of ML optimization problems. Other successful application of BCD to ML include for example the coordinate descent algorithm for LASSO [212], and the SMO algorithm for SVM [213]. The EM algorithm established for CHMM in this chapter follows the standard EM framework. To reduce the computational time and space cost one can just store and incrementally update the sufficient statistics, In particular in the E step dealing with the HMM, the incremental algorithms designed in [214, 215] might be helpful. Although the proposed methods are motivated by outlier detection, they are both general modeling tools that can be used for other applications involving smoothing, estimation and prediction of multiple time series.

Future work concerning PDM consists of two aspects: First of all, the ML methods and the PDM optimization procedure could be extended to many other CPS applications. For example, although this chapter is focused on a classification setting, both the CPLM and the veto-consensus learning can be extended for regression purposes by using a two-sided loss function. The learning result can be used as the objective function of some convex optimization for optimal control purposes. The HS^3M provides a framework to bridge semi-supervised learning and structured learning, which could be utilized generally

for object detection applications. Secondly, there is still room to improve the proposed PDM optimization algorithm. The level set construction procedure and the associated linear programmings can be computed in a parallel manner for acceleration. Moreover, the critical regions can be approximated [216] to reduce the number of invocations of the base quadratic solver.

Bibliography

[1] Edward A Lee. "CPS foundations". In: *Design Automation Conference (DAC), 2010 47th ACM/IEEE*. IEEE. 2010, pp. 737–742.

[2] Jianhua Shi, Jiafu Wan, Hehua Yan, and Hui Suo. "A survey of cyber-physical systems". In: *Wireless Communications and Signal Processing (WCSP), 2011 International Conference on*. IEEE. 2011, pp. 1–6.

[3] Patricia Derler, Edward A Lee, and Alberto Sangiovanni Vincentelli. "Modeling cyber–physical systems". In: *Proceedings of the IEEE* 100.1 (2012), pp. 13–28.

[4] Sang C Suh, U John Tanik, John N Carbone, and Abdullah Eroglu. "Applied cyber-physical systems". In: *Springer* 2 (2014), p. 27.

[5] Siddhartha Kumar Khaitan and James D McCalley. "Design techniques and applications of cyberphysical systems: A survey". In: *IEEE Systems Journal* 9.2 (2015), pp. 350–365.

[6] Ming Jin, Lillian J. Ratliff, Ioannis C. Konstantakopoulos, Costas Spanos, and Shankar Sastry. "REST: A Reliable Estimation and Stopping Time Algorithm for Social Game Experiments". In: *The 6th International Conference on Cyber-Physical Systems*. 2015, pp. 90–99.

[7] Kevin Weekly et al. "Building-in-Briefcase (BiB)". In: *arXiv preprint arXiv:1409.1660* (2014).

[8] Ming Jin et al. "Environmental Sensing by Wearable Device for Indoor Activity and Location Estimation". In: *40th Annual Conference of the IEEE Industrial Electronics Society (IECON 2014)*. 2014, pp. 5369–5375.

[9] Ming Jin, Nikolaos Bekiaris-Liberis, Kevin Weekly, Costas Spanos, and Alexandre Bayen. "Sensing by proxy: Occupancy detection based on indoor CO2 concentration". In: *UBICOMM 2015* (2015), p. 14.

[10] Kevin Weekly, Nikolaos Bekiaris-Liberis, Ming Jin, and Alexandre M Bayen. "Modeling and Estimation of the Humans' Effect on the CO2 Dynamics Inside a Conference Room". In: *IEEE Transactions on Control Systems Technology* (2015), pp. 1770–1781.

[11] A Allouhi et al. "Energy consumption and efficiency in buildings: current status and future trends". In: *Journal of Cleaner Production* 109 (2015), pp. 118–130.

[12] Ming Jin, Nikos Bekiaris-Liberis, Kevin Weekly, Costas Spanos, and Alex Bayen. "Occupancy detection via environmental sensing". In: *Transaction on Automation Science Engineering* 99 (2016), pp. 1–13.

[13] Ming Jin, Ruoxi Jia, Zhaoyi Kang, Ioannis C Konstantakopoulos, and Costas J Spanos. "Presencesense: Zero-training algorithm for individual presence detection based on power monitoring". In: *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*. ACM. 2014, pp. 1–10.

[14] Ming Jin, Wei Feng, Ping Liu, Chris Marnay, and Costas Spanos. "MOD-DR: Microgrid optimal dispatch with demand response". In: *Applied Energy* 187 (2017), pp. 758–776.

[15] Richard Brown. "Using wireless power meters to measure energy use of miscellaneous and electronic devices in buildings". In: *Energy Efficiency in Domestic Appliances and Lighting (EEDAL) 2011 Conference, Copenhagen, Denmark, May 24-26, 2011*. 2012.

[16] Zhaoyi Kang, Yuxun Zhou, Lin Zhang, and Costas J Spanos. "Virtual power sensing based on a multiple-hypothesis sequential test". In: *Smart Grid Communications (SmartGridComm), 2013 IEEE International Conference on*. IEEE. 2013, pp. 785–790.

[17] Han Zou, Lihua Xie, Qing-Shan Jia, and Hengtao Wang. "Platform and algorithm development for a rfid-based indoor positioning system". In: *Unmanned Systems* 2.03 (2014), pp. 279–291.

[18] Kevin Weekly, Han Zou, Lihua Xie, Qing-Shan Jia, and Alexandre M Bayen. "Indoor occupant positioning system using active RFID deployment and particle filters". In: *2014 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE. 2014, pp. 35–42.

[19] Han Zou et al. "BlueDetect: An iBeacon-Enabled Scheme for Accurate and Energy-Efficient Indoor-Outdoor Detection and Seamless Location-Based Service". In: *Sensors* (2016).

[20] Xiaoxuan Lu, Han Zou, Hongming Zhou, Lihua Xie, and Guang-Bin Huang. "Robust extreme learning machine with its application to indoor positioning". In: *IEEE transactions on cybernetics* 46.1 (2016), pp. 194–205.

[21] Han Zou et al. "A transfer kernel learning based strategy for adaptive localization in dynamic indoor environments: poster". In: *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM. 2016, pp. 462–464.

[22] Ruoxi Jia et al. "MapSentinel: Can the Knowledge of Space Use Improve Indoor Tracking Further?" In: *Sensors* 16.4 (2016), p. 472.

[23] Han Zou, Ming Jin, Hao Jiang, Lihua Xie, and Costas Spanos. "WinIPS: WiFi-based non-intrusive IPS for online radio map construction". In: *IEEE Transactions on Wireless Communications* ().

[24] Steven T Bushby. "BACnet Today: Significant new features and future enhancements". In: *ASHRAE journal* 44.10 (2002), S10.

[25] Ruoxi Jia, Ming Jin, Zilong Chen, and Costas Spanos. "SoundLoc: Accurate Room-level Indoor Localization using Acoustic Signatures". In: *IEEE International Conference on Automation Science and Engineering (IEEE CASE 2015)*. 2015, pp. 186–193.

[26] Ming Jin and Costas Spanos. "BRIEF: Bayesian Regression of Infinite Expert Forecasters for Single and Multiple Time Series Prediction". In: *54th IEEE Conference on Decision and Control (CDC 2015)*. 2015, pp. 78–83.

[27] Ming Jin, Lin Zhang, and Costas Spanos. "Power Prediction through Energy Consumption Pattern Recognition for Smart Buildings". In: *IEEE International Conference on Automation Science and Engineering (IEEE CASE 2015)*. 2015, pp. 419–424.

[28] Zhaoyi Kang and Costas J Spanos. "Sequential logistic principal component analysis (SLPCA): Dimensional reduction in streaming multivariate binary-state system". In: *Machine Learning and Applications (ICMLA), 2014 13th International Conference on*. IEEE. 2014, pp. 171–177.

[29] Yuxun Zhou, Zhaoyi Kang, Lin Zhang, and Costas Spanos. "Causal analysis for non-stationary time series in sensor-rich smart buildings". In: *Automation Science and Engineering (CASE), 2013 IEEE International Conference on*. IEEE. 2013, pp. 593–598.

[30] Yuxun Zhou and Costas J Spanos. "Causal meets Submodular: Subset Selection with Directed Information". In: *Advances In Neural Information Processing Systems*. 2016, pp. 2649–2657.

[31] Yuxun Zhou, Dan Li, and Costas J Spanos. "Learning optimization friendly comfort model for hvac model predictive control". In: *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*. IEEE. 2015, pp. 430–439.

[32] Kevin Weekly et al. "Low-cost coarse airborne particulate matter sensing for indoor occupancy detection". In: *Automation Science and Engineering (CASE), 2013 IEEE International Conference on*. IEEE. 2013, pp. 32–37.

[33] Ming Jin, Ruoxi Jia, and Costas Spanos. "Virtual Occupancy Sensing: Using Smart Meters to Indicate Your Presence". In: *IEEE Transactions on Mobile Computing* (2017).

[34] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer Science & Business Media, 2013.

[35] Han Zou et al. "WinLight: An WiFi-based Occupancy Adaptive Lighting Control System for Smart Building". In: *Energy and Building*. Elsevier. 2017.

[36] Han Zou, Hengtao Wang, Lihua Xie, and Qing-Shan Jia. "An RFID indoor positioning system by using weighted path loss and extreme learning machine". In: *2013 IEEE 1st International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA)*. IEEE. 2013, pp. 66–71.

[37] Han Zou, Lihua Xie, Qing-Shan Jia, and Hengtao Wang. "An integrative weighted path loss and extreme learning machine approach to rfid based indoor positioning". In: *2013 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE. 2013, pp. 1–5.

[38] Zhenghua Chen et al. "Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization". In: *Sensors* 15.1 (2015), pp. 715–732.

[39] Han Zou, Zhenghua Chen, Hao Jiang, Lihua Xie, and Costas Spanos. "Accurate Indoor Localization and Tracking Using Mobile Phone Inertial Sensors, WiFi and iBeacon". In: *2017 IEEE International Symposium on Inertial Sensors and Systems*. IEEE. 2017.

[40] Baoqi Huang, Guodong Qi, Xiaokun Yang, Long Zhao, and Han Zou. "Exploiting cyclic features of walking for pedestrian dead reckoning with unconstrained smartphones". In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*. ACM. 2016, pp. 374–385.

[41] Han Zou, Baoqi Huang, Xiaoxuan Lu, Hao Jiang, and Lihua Xie. "A robust indoor positioning system based on the procrustes analysis and weighted extreme learning machine". In: *IEEE Transactions on Wireless Communications* 15.2 (2016), pp. 1252–1266.

[42] Dimitrios Lymberopoulos et al. "A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned". In: *Proceedings of the 14th international conference on information processing in sensor networks*. ACM. 2015, pp. 178–189.

[43] Han Zou, Ming Jin, Hao Jiang, Lihua Xie, and Costas Spanos. "WinIPS: WiFi-based non-intrusive IPS for online radio map construction". In: *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE. 2016, pp. 1081–1082.

[44] Xiaoxuan Lu, Chengpu Yu, Han Zou, Hao Jiang, and Lihua Xie. "Extreme learning machine with dead zone and its application to WiFi based indoor positioning". In: *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*. IEEE. 2014, pp. 625–630.

[45] Han Zou, Yiwen Luo, Xiaoxuan Lu, Hao Jiang, and Lihua Xie. "A mutual information based online access point selection strategy for WiFi indoor localization". In: *2015 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE. 2015, pp. 180–185.

[46] Han Zou, Xiaoxuan Lu, Hao Jiang, and Lihua Xie. "A fast and precise indoor localization algorithm based on an online sequential extreme learning machine". In: *Sensors* 15.1 (2015), pp. 1804–1824.

[47] Clive WJ Granger. "Investigating causal relations by econometric models and cross-spectral methods". In: *Econometrica: Journal of the Econometric Society* (1969), pp. 424–438.

[48] Judea Pearl. *Causality: Models, Reasoning and Inference (Second Edition)*. Cambridge university press, 2009.

[49] Aurelie C Lozano et al. "Spatial-temporal causal modeling for climate change attribution". In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2009, pp. 587–596.

[50] Mohammad Taha Bahadori and Yan Liu. "An examination of practical granger causality inference". In: *Proceedings of the 2013 SIAM International Conference on data Mining*. SIAM. 2013, pp. 467–475.

[51] Kenneth A Bollen. *Structural equations with latent variables*. John Wiley and Sons, 2014.

[52] Shohei Shimizu and Yutaka Kano. "Use of non-normality in structural equation modeling: Application to direction of causation". In: *Journal of Statistical Planning and Inference* 138.11 (2008), pp. 3483–3491.

[53] Pramod Mathai, Nuno C Martins, and Benjamin Shapiro. "On the detection of gene network interconnections using directed mutual information". In: *Information Theory and Applications Workshop, 2007*. IEEE. 2007, pp. 274–283.

[54] Christopher J Quinn, Todd P Coleman, Negar Kiyavash, and Nicholas G Hatsopoulos. "Estimating the directed information to infer causal relationships in ensemble neural spike train recordings". In: *Journal of computational neuroscience* 30.1 (2011), pp. 17–44.

[55] Greg Ver Steeg and Aram Galstyan. "Information-theoretic measures of influence based on content dynamics". In: *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM. 2013, pp. 3–12.

[56] Nuala A Sheehan, Vanessa Didelez, Paul R Burton, and Martin D Tobin. "Mendelian randomisation and causal inference in observational epidemiology". In: *PLoS Med* 5.8 (2008), e177.

[57] Christopher J Quinn, Negar Kiyavash, and Todd P Coleman. "Directed information graphs". In: *Information Theory, IEEE Transactions on* 61.12 (2015), pp. 6887–6909.

[58] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. "An analysis of approximations for maximizing submodular set functions I". In: *Mathematical Programming* 14.1 (1978), pp. 265–294.

[59] Marshall L Fisher, George L Nemhauser, and Laurence A Wolsey. "An analysis of approximations for maximizing submodular set functions II". In: *Polyhedral combinatorics*. Springer, 1978, pp. 73–87.

[60] Andreas Krause and Carlos E Guestrin. "Near-optimal nonmyopic value of information in graphical models". In: *arXiv preprint arXiv:1207.1394* (2012).

[61] Andreas Krause, Ajit Singh, and Carlos Guestrin. "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies". In: *Journal of Machine Learning Research* 9.Feb (2008), pp. 235–284.

[62] Hui Lin and Jeff Bilmes. "Multi-document summarization via budgeted maximization of submodular functions". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics. 2010, pp. 912–920.

[63] Hui Lin and Jeff Bilmes. "A class of submodular functions for document summarization". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics. 2011, pp. 510–520.

[64] Abhimanyu Das and David Kempe. "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection". In: *arXiv preprint arXiv:1102.3975* (2011).

[65] Zoë Abrams, Ashish Goel, and Serge Plotkin. "Set k-cover algorithms for energy efficient monitoring in wireless sensor networks". In: *Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM. 2004, pp. 424–432.

[66] Andreas Krause and Daniel Golovin. "Submodular function maximization". In: *Tractability: Practical Approaches to Hard Problems* 3 (2012), p. 19.

[67] Alexander Schrijver. "A combinatorial algorithm minimizing submodular functions in strongly polynomial time". In: *Journal of Combinatorial Theory, Series B* 80.2 (2000), pp. 346–355.

[68] Chun-Wa Ko, Jon Lee, and Maurice Queyranne. "An exact algorithm for maximum entropy sampling". In: *Operations Research* 43.4 (1995), pp. 684–691.

[69] Uriel Feige, Vahab S Mirrokni, and Jan Vondrak. "Maximizing non-monotone submodular functions". In: *SIAM Journal on Computing* 40.4 (2011), pp. 1133–1153.

[70] Moran Feldman, Joseph Naor, and Roy Schwartz. "A unified continuous greedy algorithm for submodular maximization". In: *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*. IEEE. 2011, pp. 570–579.

[71] Niv Buchbinder, Moran Feldman, Joseph Seffi Naor, and Roy Schwartz. "Submodular maximization with cardinality constraints". In: *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics. 2014, pp. 1433–1452.

[72] Volkan Cevher and Andreas Krause. "Greedy dictionary selection for sparse representation". In: *Selected Topics in Signal Processing, IEEE Journal of* 5.5 (2011), pp. 979–988.

[73] Yoshinobu Kawahara and Takashi Washio. "Prismatic algorithm for discrete DC programming problem". In: *Advances in Neural Information Processing Systems*. 2011, pp. 2106–2114.

[74] Mukund Narasimhan and Jeff A Bilmes. "A submodular-supermodular procedure with applications to discriminative structure learning". In: *arXiv preprint arXiv:1207.1404* (2012).

[75] Donatello Materassi and Giacomo Innocenti. "Topological identification in networks of dynamical systems". In: *Automatic Control, IEEE Transactions on* 55.8 (2010), pp. 1860–1871.

[76] Andrew Bolstad, Barry D Van Veen, and Robert Nowak. "Causal network inference via group sparse regularization". In: *Signal Processing, IEEE Transactions on* 59.6 (2011), pp. 2628–2641.

[77] Sahand Negahban, Bin Yu, Martin J Wainwright, and Pradeep K Ravikumar. "A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers". In: *Advances in Neural Information Processing Systems*. 2009, pp. 1348–1356.

[78] Jiantao Jiao, Haim H Permuter, Lei Zhao, Young-Han Kim, and Tsachy Weissman. "Universal estimation of directed information". In: *Information Theory, IEEE Transactions on* 59.10 (2013), pp. 6220–6242.

[79] Kevin Murphy et al. "The bayes net toolbox for matlab". In: *Computing science and statistics* 33.2 (2001), pp. 1024–1034.

[80] Charu C Aggarwal and Philip S Yu. "Outlier detection with uncertain data". In: *Proceedings of the 2008 SIAM International Conference on Data Mining*. SIAM. 2008, pp. 483–493.

[81] Charu C Aggarwal and Philip S Yu. "Outlier detection for high dimensional data". In: *ACM Sigmod Record*. Vol. 30. 2. ACM. 2001, pp. 37–46.

[82] Manish Gupta, Jing Gao, Charu C Aggarwal, and Jiawei Han. "Outlier detection for temporal data: A survey". In: *IEEE Transactions on Knowledge and Data Engineering* 26.9 (2014), pp. 2250–2267.

[83] Charu C Aggarwal, Yuchen Zhao, and S Yu Philip. "Outlier detection in graph streams". In: *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*. IEEE. 2011, pp. 399–409.

[84] Manish Gupta, Jing Gao, Yizhou Sun, and Jiawei Han. "Integrating community matching and outlier detection for mining evolutionary community outliers". In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2012, pp. 859–867.

[85] Charu C Aggarwal. "Outlier analysis". In: *Data mining*. Springer. 2015, pp. 237–263.

[86] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey". In: *ACM computing surveys (CSUR)* 41.3 (2009), p. 15.

[87] Victoria J Hodge and Jim Austin. "A survey of outlier detection methodologies". In: *Artificial intelligence review* 22.2 (2004), pp. 85–126.

[88] Dan Li, Yuxun Zhou, Guoqiang Hu, and Costas J Spanos. "Fault detection and diagnosis for building cooling system with a tree-structured learning method". In: *Energy and Buildings* 127 (2016), pp. 540–551.

[89] Yuxun Zhou, Reza Arghandeh, Ioannis Konstantakopoulos, Shayaan Abdullah, and Costas J Spanos. "Data-driven event detection with partial knowledge: A hidden structure semi-supervised learning method". In: *American Control Conference (ACC), 2016*. IEEE. 2016, pp. 5962–5968.

[90] Yuxun Zhou, Reza Arghandeh, and Costas J Spanos. "Partial Knowledge Data-driven Event Detection for Power Distribution Networks". In: *IEEE Transactions on Smart Grid* (2017).

[91] Han Zou, Hao Jiang, Xiaoxuan Lu, and Lihua Xie. "An online sequential extreme learning machine approach to WiFi based indoor positioning". In: *2014 IEEE World Forum on Internet of Things (WF-IoT)*. IEEE. 2014, pp. 111–116.

[92] Xiaoxuan Lu, Yushen Long, Han Zou, Yu Chengpu, and Lihua Xie. "Robust extreme learning machine for regression problems with its application to wifi based indoor positioning system". In: *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE. 2014, pp. 1–6.

[93] Han Zou, Baoqi Huang, Xiaoxuan Lu, Hao Jiang, and Lihua Xie. "Standardizing location fingerprints across heterogeneous mobile devices for indoor localization". In: *2016 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE. 2016, pp. 1–6.

[94] Han Zou, Zhirong Qiu, Hao Jiang, Lihua Xie, and Yiguang Hong. "Consensus-Based Parallel Extreme Learning Machine for Indoor Localization". In: *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2016, pp. 1–6.

[95] Han Zou et al. "Adaptive Localization in Dynamic Indoor Environments by Transfer Kernel Learning". In: *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE. 2017.

[96] Han Zou, Yuxun Zhou, Jianfei Yang, Lihua Xie, and Costas Spanos. "Non-intrusive Occupancy Sensing in Commercial Buildings". In: *Energy and Buildings* (2017).

[97] Han Zou, Yuxun Zhou, Jianfei Yang, Lihua Xie, and Costas Spanos. "FreeDetector: Device-Free Occupancy Detection with Commodity WiFi". In: *2017 IEEE SECON Workshop on Smart and Connected Indoor Environments (SCIE)*. IEEE. 2017.

[98] Han Zou, Yuxun Zhou, Jianfei Yang, Lihua Xie, and Costas Spanos. "FreeCount: Device-Free Crowd Counting with Commodity WiFi". In: *2017 IEEE Global Communications Conference (GLOBECOM)*. IEEE. 2017.

[99] Rolf Isermann. "Model-based fault-detection and diagnosis–status and applications". In: *Annual Reviews in control* 29.1 (2005), pp. 71–85.

[100] Borhan M Sanandaji, Eilyan Bitar, Kameshwar Poolla, and Tyrone L Vincent. "An abrupt change detection heuristic with applications to cyber data attacks on power systems". In: *American Control Conference (ACC), 2014*. IEEE. 2014, pp. 5056–5061.

[101] Peyman Mohajerin Esfahani, Maria Vrakopoulou, Kostas Margellos, John Lygeros, and Göran Andersson. "Cyber attack in a two-area power system: Impact identification using reachability". In: *American Control Conference (ACC), 2010*. IEEE. 2010, pp. 962–967.

[102] Xuemei Ding, Josiah Poon, Ivan Celanovic, and Alejandro D Dominguez-Garcia. "Fault detection and isolation filters for three-phase ac-dc power electronics systems". In: *Circuits and Systems I: Regular Papers, IEEE Transactions on* 60.4 (2013), pp. 1038–1051.

[103] Anselm Schwarte, Frank Kimmidi, and Rolf Isermann. "Model-based fault detection of a diesel engine with turbo charger-a case study". In: *Fault Detection, Supervision and Safety of Technical Processes 2003 (SAFEPROCESS 2003): A Proceedings Volume from the 5th IFAC Symposium, Washington, DC, USA, 9-11 June 2003*. Vol. 1. Elsevier. 2004, p. 293.

[104] G Cavraro, R Arghandeh, G Barchi, and A von Meier. "Distribution network topology detection with time-series measurements". In: *Innovative Smart Grid Technologies Conference (ISGT), 2015 IEEE Power & Energy Society*. IEEE. 2015, pp. 1–5.

[105] Guido Cavraro, Reza Arghandeh, Alexandra von Meier, and Kameshwar Poolla. "Data-Driven Approach for Distribution Network Topology Detection". In: *arXiv preprint arXiv:1504.00724* (2015).

[106] Wen Jiang, Matthew L Baker, Qiu Wu, Chandrajit Bajaj, and Wah Chiu. "Applications of a bilateral denoising filter in biological electron microscopy". In: *Journal of structural biology* 144.1 (2003), pp. 114–122.

[107] Stephane Mallat. *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.

[108] David L Donoho and Iain M Johnstone. "Adapting to unknown smoothness via wavelet shrinkage". In: *Journal of the american statistical association* 90.432 (1995), pp. 1200–1224.

[109] S Joe Qin. "Survey on data-driven industrial process monitoring and diagnosis". In: *Annual Reviews in Control* 36.2 (2012), pp. 220–234.

[110] W Enders. "Applied Econometric Time Series, by Walter". In: *Technometrics* 46.2 (2004), p. 264.

[111] Jianqing Fan and Irene Gijbels. *Local polynomial modelling and its applications: monographs on statistics and applied probability 66*. Vol. 66. CRC Press, 1996.

[112] Robert J Hodrick and Edward C Prescott. "Postwar US business cycles: an empirical investigation". In: *Journal of Money, credit, and Banking* (1997), pp. 1–16.

[113] Samprit Chatterjee and Ali S Hadi. *Regression analysis by example*. John Wiley & Sons, 2015.

[114] Ray M Bowen and Chao-Cheng Wang. *Introduction to vectors and tensors*. Vol. 2. Courier Corporation, 2008.

[115] Jürgen Forster and Manfred K Warmuth. "Relative expected instantaneous loss bounds". In: *Journal of Computer and System Sciences* 64.1 (2002), pp. 76–102.

[116] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[117] Dimitri P Bertsekas, Angelia Nedi, Asuman E Ozdaglar, et al. "Convex analysis and optimization". In: (2003).

[118] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. "Distributed optimization and statistical learning via the alternating direction method of multipliers". In: *Foundations and Trends® in Machine Learning* 3.1 (2011), pp. 1–122.

[119] André R Gonçalves, Fernando J Von Zuben, and Arindam Banerjee. "Multi-task sparse structure learning with Gaussian copula models". In: *Journal of Machine Learning Research* 17.33 (2016), pp. 1–30.

[120] Dheeraj Bokde, Sheetal Girase, and Debajyoti Mukhopadhyay. "Matrix factorization model in collaborative filtering algorithms: A survey". In: *Procedia Computer Science* 49 (2015), pp. 136–146.

[121] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[122] Rudolph Emil Kalman et al. "A new approach to linear filtering and prediction problems". In: (1960).

[123] Eduardo D Sontag. *Mathematical control theory: deterministic finite dimensional systems*. Vol. 6. Springer Science & Business Media, 2013.

[124] Pingping Zhu, Badong Chen, and Jose C Principe. "Learning nonlinear generative models of time series with a Kalman filter in RKHS". In: *IEEE Transactions on Signal Processing* 62.1 (2014), pp. 141–155.

[125] Rickard Karlsson, Thomas Schon, and Fredrik Gustafsson. "Complexity analysis of the marginalized particle filter". In: *IEEE Transactions on Signal Processing* 53.11 (2005), pp. 4408–4411.

[126] CF Jeff Wu. "On the convergence properties of the EM algorithm". In: *The Annals of statistics* (1983), pp. 95–103.

[127] Richard A Redner and Homer F Walker. "Mixture densities, maximum likelihood and the EM algorithm". In: *SIAM review* 26.2 (1984), pp. 195–239.

[128] Xiao-Li Meng and Donald B Rubin. "On the global and componentwise rates of convergence of the EM algorithm". In: *Linear Algebra and its Applications* 199 (1994), pp. 413–425.

[129] Chong Wu, Can Yang, Hongyu Zhao, and Ji Zhu. "On the Convergence of the EM Algorithm: From the Statistical Perspective". In: *arXiv preprint arXiv:1611.00519* (2016).

[130] Geoffrey McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*. Vol. 382. John Wiley & Sons, 2007.

[131] Yuxun Zhou, Reza Arghandeh, and Costas J Spanos. "Online learning of Contextual Hidden Markov Models for temporal-spatial data analysis". In: *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE. 2016, pp. 6335–6341.

[132] A. Von Meier, D. Culler, A. McEachern, and R. Arghandeh. "Micro-synchrophasors for distribution systems". In: *Innovative Smart Grid Technologies Conference (ISGT), 2014 IEEE PES*. Feb. 2014, pp. 1–5.

[133] Bruno Sinopoli et al. "Kalman filtering with intermittent observations". In: *IEEE transactions on Automatic Control* 49.9 (2004), pp. 1453–1464.

[134] Chong Gu. *Smoothing spline ANOVA models*. Vol. 297. Springer Science & Business Media, 2013.

[135] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[136] Kemin Zhou, John Comstock Doyle, Keith Glover, et al. *Robust and optimal control*. Vol. 40. Prentice hall New Jersey, 1996.

[137] Michael Athans and Peter L Falb. *Optimal control: an introduction to the theory and its applications*. Courier Corporation, 2013.

[138] Pieter Eykhoff. *Trends and progress in system identification: IFAC Series for Graduates, Research Workers & Practising Engineers*. Vol. 1. Elsevier, 2014.

[139] Oliver Nelles. *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*. Springer Science & Business Media, 2013.

[140] Mehdi Maasoumy, Alessandro Pinto, and Alberto Sangiovanni-Vincentelli. "Model-based hierarchical optimal control design for HVAC systems". In: *ASME 2011 Dynamic Systems and Control Conference and Bath/ASME Symposium on Fluid Power and Motion Control*. American Society of Mechanical Engineers. 2011, pp. 271–278.

[141] Biao Sun et al. "Building energy management: Integrated control of active and passive heating, cooling, lighting, shading, and ventilation systems". In: *IEEE Transactions on automation science and engineering* 10.3 (2013), pp. 588–602.

[142] Dimitri P Bertsekas, Dimitri P Bertsekas, Dimitri P Bertsekas, and Dimitri P Bertsekas. *Dynamic programming and optimal control*. Vol. 1. 2. Athena Scientific Belmont, MA, 1995.

[143] M Murat Dundar, Matthias Wolf, Sarang Lakare, Marcos Salganicoff, and Vikas C Raykar. "Polyhedral classifier for target detection: a case study: colorectal cancer". In: *Proceedings of the 25th international conference on Machine learning*. ACM. 2008, pp. 288–295.

[144] Naresh Manwani and PS Sastry. "Learning Polyhedral Classifiers Using Logistic Function." In: *ACML*. 2010, pp. 17–30.

[145] Naresh Manwani and PS Sastry. "Polyceptron: A polyhedral learning algorithm". In: *arXiv preprint arXiv:1107.1564* (2011).

[146] Alex Kantchelian et al. "Large-margin convex polytope machine". In: *Advances in Neural Information Processing Systems*. 2014, pp. 3248–3256.

[147] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[148] Hamed Masnadi-Shirazi and Nuno Vasconcelos. "Risk minimization, probability elicitation, and cost-sensitive SVMs." In: *ICML*. 2010, pp. 759–766.

[149] Yiming Ying and Colin Campbell. "Generalization bounds for learning the kernel". In: *22nd Annual Conference on Learning Theory (COLT 2009)*. 2009.

[150] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. "Generalization bounds for learning kernels". In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. 2010, pp. 247–254.

[151] Zakria Hussain and John Shawe-Taylor. "Improved loss bounds for multiple kernel learning". In: *International Conference on Artificial Intelligence and Statistics*. 2011, pp. 370–377.

[152] Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. "Learning the kernel matrix with semidefinite programming". In: *The Journal of Machine Learning Research* 5 (2004), pp. 27–72.

[153] Alain Rakotomamonjy, Francis Bach, Stéphane Canu, and Yves Grandvalet. "SimpleMKL". In: *Journal of Machine Learning Research* 9 (2008), pp. 2491–2521.

[154] Marius Kloft, Ulf Brefeld, Sören Sonnenburg, and Alexander Zien. "Lp-norm multiple kernel learning". In: *The Journal of Machine Learning Research* 12 (2011), pp. 953–997.

[155] Saketha N Jagarlapudi et al. "On the algorithmics and applications of a mixed-norm based kernel learning formulation". In: *Advances in neural information processing systems*. 2009, pp. 844–852.

[156] Xinwang Liu, Lei Wang, Jian Zhang, and Jianping Yin. "Sample-adaptive multiple kernel learning". In: *Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI-14)* (2014).

[157] Xinwang Liu, Lei Wang, Jianping Yin, Yong Dou, and Jian Zhang. "Absent Multiple Kernel Learning". In: *Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)* (2015).

[158] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. "Learning non-linear combinations of kernels". In: *Advances in neural information processing systems*. 2009, pp. 396–404.

[159] Francis Bach. "High-dimensional non-linear variable selection through hierarchical kernel learning". In: *arXiv preprint arXiv:0909.0844* (2009).

[160] Cheng S Ong, Robert C Williamson, and Alex J Smola. "Learning the kernel with hyperkernels". In: *Journal of Machine Learning Research*. 2005, pp. 1043–1071.

[161] Bernhard Schölkopf, Alex J Smola, Robert C Williamson, and Peter L Bartlett. "New support vector algorithms". In: *Neural computation* 12.5 (2000), pp. 1207–1245.

[162] Xiaojin Zhu. "Semi-supervised learning". In: *Encyclopedia of Machine Learning*. Springer, 2011, pp. 892–897.

[163] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]". In: *IEEE Transactions on Neural Networks* 20.3 (2009), pp. 542–542.

[164] Xiaojin Zhu. "Semi-supervised learning literature survey". In: (2005).

[165] Yuxun Zhou, Zhaoyi Kang, and Costas J Spanos. "Parametric dual maximization for non-convex learning problems". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*. AAAI Press. 2017.

[166] Kristin Bennett, Ayhan Demiriz, et al. "Semi-supervised support vector machines". In: *Advances in Neural Information processing systems* (1999), pp. 368–374.

[167] Thorsten Joachims. "Transductive inference for text classification using support vector machines". In: *ICML*. Vol. 99. 1999, pp. 200–209.

[168] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. "Object detection with discriminatively trained part-based models". In: *PAMI, IEEE Transactions on* 32.9 (2010), pp. 1627–1645.

[169] Chun-Nam John Yu and Thorsten Joachims. "Learning structural SVMs with latent variables". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM. 2009, pp. 1169–1176.

[170] Yuxun Zhou, Ninghang Hu, and Costas J Spanos. "Veto-consensus multiple kernel learning". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press. 2016, pp. 2407–2414.

[171] Linli Xu, Koby Crammer, and Dale Schuurmans. "Robust support vector machine training via convex outlier ablation". In: *AAAI*. Vol. 6. 2006, pp. 536–542.

[172] Alan L Yuille, Anand Rangarajan, and AL Yuille. "The concave-convex procedure (CCCP)". In: *Advances in neural information processing systems* 2 (2002), pp. 1033–1040.

[173] Wei Ping, Qiang Liu, and Alexander Ihler. "Marginal Structured SVM with Hidden Variables". In: *arXiv preprint arXiv:1409.1320* (2014).

[174] Léon Bottou. "Large-scale machine learning with stochastic gradient descent". In: *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[175] Yuxun Zhou, Baihong Jin, and Costas J Spanos. "Learning convex piecewise linear machine for data-driven optimal control". In: *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2015, pp. 966–972.

[176] Olivier Chapelle and Alexander Zien. "Semi-supervised classification by low density separation". In: *Proceedings of the tenth international workshop on artificial intelligence and statistics*. Vol. 1. 2005, pp. 57–64.

[177] Olivier Chapelle, Vikas Sindhwani, and S Sathiya Keerthi. "Branch and bound for semi-supervised support vector machines". In: *Advances in neural information processing systems*. 2006, pp. 217–224.

[178] Olivier Chapelle, Mingmin Chi, and Alexander Zien. "A continuation method for semi-supervised SVMs". In: *Proceedings of the 23rd international conference on Machine learning*. ACM. 2006, pp. 185–192.

[179] T.De Bei and N Cristianini. "Semi-supervised learning using semi-definite programming". In: *Semi-supervised Learning*. MIT Press, 2006, pp. 177–186.

[180] Jaehyun Park and Stephen Boyd. "A Semidefinite Programming Method for Integer Convex Quadratic Minimization". In: *arXiv preprint arXiv:1504.07672* (2015).

[181] Bala Krishnamoorthy. "Bounds on the size of branch-and-bound proofs for integer knapsacks". In: *Operations Research Letters* 36.1 (2008), pp. 19–25.

[182] Olivier Chapelle, Vikas Sindhwani, and Sathiya S Keerthi. "Optimization techniques for semi-supervised support vector machines". In: *The Journal of Machine Learning Research* 9 (2008), pp. 203–233.

[183] Petter TøNdel, Tor Arne Johansen, and Alberto Bemporad. "An algorithm for multi-parametric quadratic programming and explicit MPC solutions". In: *Automatica* 39.3 (2003), pp. 489–497.

[184] Gerd Wachsmuth. "On LICQ and the uniqueness of Lagrange multipliers". In: *Operations Research Letters* 41.1 (2013), pp. 78–80.

[185] Trevor Hastie, Saharon Rosset, Robert Tibshirani, and Ji Zhu. "The entire regularization path for the support vector machine". In: *The Journal of Machine Learning Research* 5 (2004), pp. 1391–1415.

[186] Masayuki Karasuyama and Ichiro Takeuchi. "Suboptimal solution path algorithm for support vector machine". In: *ICML* (2011).

[187] Ider Tsevendorj. "Piecewise-convex maximization problems". In: *Journal of Global Optimization* 21.1 (2001), pp. 1–14.

[188] Pando G Georgiev, Altannar Chinchuluun, and Panos M Pardalos. "Optimality conditions of first order for global minima of locally Lipschitz functions". In: *Optimization* 60.1-2 (2011), pp. 277–282.

[189] Chih-Chung Chang and Chih-Jen Lin. "LIBSVM: a library for support vector machines". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 2.3 (2011), p. 27.

[190] Ronan Collobert, Fabian Sinz, Jason Weston, and Léon Bottou. "Large scale transductive SVMs". In: *The Journal of Machine Learning Research* 7 (2006), pp. 1687–1712.

[191] Vikas Sindhwani, S Sathiya Keerthi, and Olivier Chapelle. "Deterministic annealing for semi-supervised kernel machines". In: *Proceedings of the 23rd international conference on Machine learning*. ACM. 2006, pp. 841–848.

[192] Kohei Ogawa, Motoki Imamura, Ichiro Takeuchi, and Masashi Sugiyama. "Infinitesimal annealing for training semi-supervised support vector machines". In: *Proceedings of the 30th International Conference on Machine Learning*. 2013, pp. 897–905.

[193] Abdul Afram and Farrokh Janabi-Sharifi. "Theory and applications of HVAC control systems–A review of model predictive control (MPC)". In: *Building and Environment* 72 (2014), pp. 343–355.

[194] YH Yau and BT Chew. "A review on predicted mean vote and adaptive thermal comfort models". In: *Building Services Engineering Research and Technology* 35.1 (2014), pp. 23–35.

[195] M Castilla, JD Álvarez, MG Ortega, and MR Arahal. "Neural network and polynomial approximated thermal comfort models for HVAC systems". In: *Building and Environment* 59 (2013), pp. 107–115.

[196] Abdul Afram, Farrokh Janabi-Sharifi, Alan S Fung, and Kaamran Raahemifar. "Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system". In: *Energy and Buildings* 141 (2017), pp. 96–113.

[197] Ahmed Cherif Megri and Issam El Naqa. "Prediction of the thermal comfort indices using improved support vector machine classifiers and nonlinear kernel functions". In: *Indoor and Built Environment* 25.1 (2016), pp. 6–16.

[198] Sun Bin and Han Ke. "Indoor thermal comfort pmv index prediction based on particle swarm algorithm and least square support vector machine". In: *Intelligent System Design and Engineering Application (ISDEA), 2010 International Conference on*. Vol. 1. IEEE. 2010, pp. 857–860.

[199] Carlo Manna, Nic Wilson, and Kenneth N Brown. "Learning Individual Thermal Comfort using Robust Locally Weighted Regression with Adaptive Bandwidth". In: *Workshop on AI Problems and Approaches for Intelligent Environments*. 2012, p. 35.

[200] Lam Abraham Hang-yat and Dan Wang. "Carrying my environment with me: A participatory-sensing approach to enhance thermal comfort". In: *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*. ACM. 2013, pp. 1–8.

[201] Varick L Erickson and Alberto E Cerpa. "Thermovote: participatory sensing for efficient building hvac conditioning". In: *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*. ACM. 2012, pp. 9–16.

[202] Jay Taneja, Andrew Krioukov, Stephen Dawson-Haggerty, and David Culler. "Enabling advanced environmental conditioning with a building application stack". In: *Green Computing Conference (IGCC), 2013 International*. IEEE. 2013, pp. 1–10.

[203] Tyler Hoyt, Stefano Schiavon, Alberto Piccioli, Dustin Moon, and Kyle Steinfeld. "CBE thermal comfort tool". In: *Center for the Built Environment, University of California Berkeley, http://cbe. berkeley. edu/comforttool* (2013).

[204] Richard J De Dear, Gail Schiller Brager, James Reardon, Fergus Nicol, et al. "Developing an adaptive model of thermal comfort and preference/discussion". In: *ASHRAE transactions* 104 (1998), p. 145.

[205] Noël Djongyang, René Tchinda, and Donatien Njomo. "Thermal comfort: A review paper". In: *Renewable and Sustainable Energy Reviews* 14.9 (2010), pp. 2626–2640.

[206] Hanchuan Peng, Fuhui Long, and Chris Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy". In: *IEEE Transactions on pattern analysis and machine intelligence* 27.8 (2005), pp. 1226–1238.

[207] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10 (2010), pp. 1345–1359.

[208] Daniel Golovin and Andreas Krause. "Adaptive Submodularity: A New Approach to Active Learning and Stochastic Optimization." In: *COLT*. 2010, pp. 333–345.

[209] Daniel Golovin and Andreas Krause. "Adaptive submodularity: Theory and applications in active learning and stochastic optimization". In: *Journal of Artificial Intelligence Research* 42 (2011), pp. 427–486.

[210] Yuxin Chen et al. "Active Detection via Adaptive Submodularity." In: *ICML*. 2014, pp. 55–63.

[211] Andrew An Bian, Joachim M Buhmann, Andreas Krause, and Sebastian Tschiatschek. "Guarantees for Greedy Maximization of Non-submodular Functions with Applications". In: *arXiv preprint arXiv:1703.02100* (2017).

[212] Tong Tong Wu and Kenneth Lange. "Coordinate descent algorithms for lasso penalized regression". In: *The Annals of Applied Statistics* (2008), pp. 224–244.

[213] John Platt. "Sequential minimal optimization: A fast algorithm for training support vector machines". In: (1998).

[214] Gianluigi Mongillo and Sophie Deneve. "Online learning with hidden Markov models". In: *Neural computation* 20.7 (2008), pp. 1706–1716.

[215] Olivier Cappé. "Online sequential monte carlo em algorithm". In: *Statistical Signal Processing, 2009. SSP'09. IEEE/SP 15th Workshop on*. IEEE. 2009, pp. 37–40.

[216] Yuxun Zhou and Costas J Spanos. "On a class of multi-parametric quadratic programming and its applications to machine learning". In: *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE. 2016, pp. 2826–2833.