Title

Data augmentation for spatial disease mapping

Authors
Diniz, Raphaella
Vaz-de-Melo, Pedro
Assunção, Renato

Publication Date
2021-12-01

DOI
10.25436/E2KS35

Peer reviewed

Data augmentation for spatial disease mapping

Raphaella Carvalho Diniz[1], Pedro O.S. Vaz-de-Melo[2], and Renato Assunção[2,3]

[1] Simon Fraser University, Burnaby BC V5A 1S6, Canada raphaella_diniz@sfu.ca
[2] Universidade Federal de Minas Gerais, Belo Horizonte MG 31270-901, Brazil
{olmo, assuncao}@dcc.ufmg.br
[3] ESRI Inc., Redlands CA 92373-8100, USA

Abstract. Data augmentation is a technique to increase the amount of available data, improving the accuracy of machine learning models. Previous studies have addressed this approach in spatial data where the data is structured as points (coordinates) or surfaces. However, there are cases where the data is aggregated into adjacent polygons (e.g. cities). We introduce new methods to augment data from a single instance of a map partitioned into polygon subareas. We focus on disease mapping examples in which each polygon has a disease rate. Our methods do not change the map spatial pattern, even though the underlying distribution is unknown. Our proposal does not depend on the spatial configuration of the map, the underlying distribution of the data or other prior information.

Keywords: Cluster detection · Cluster stability · Disease cluster · Scan statistics · Spatial statistics.

DOI: https://doi.org/10.25436/E2KS35

1 Introduction

Many machine learning (ML) algorithms, mainly deep neural networks, rely on a large amount of data to avoid bias and to prevent overfitting. In typical machine learning applications, we split the dataset into training and testing samples to evaluate the results *without requiring any distributional assumptions about how the data is generated*. No need for synthetic datasets is involved in this train-test evaluation that is made application-specific and completely data-driven. This allows us to avoid overfitting and control for biased predictions.

Unfortunately, there are scenarios where data is scarce and other techniques are needed to generate more observations synthetically. Data augmentation is a largely adopted technique, especially for training neural networks with natural images [2,5]. It artificially increases the training dataset by adding slighted randomly distorted replicas of the training samples. There is an important trade-off to consider. On the one hand, the random distortion must be small to avoid destroying the intrinsic spatial correlation present in the training samples. On the other hand, it must be large to produce samples that are substantially distinct from the training examples. As a result, the machine learning model can be trained more accurately with more observations and the testing data can have enough observations to properly represent the data underlying distribution [3, 4, 7, 8].

Differently from the typical ML dataset with several independent examples, we are concerned with the data used for the spatial surveillance of disease incidence. The spatial cluster detection [6] is an epidemiological and public health task that aims at detecting geographically contiguous areas that are *anomalous* with respect to certain reference framework, usually a generative probability distribution model [1]. However, from the data mining (DM) and machine learning (ML) point of view, this is an unfortunate misnomer. What is called spatial cluster detection is the detection of this *spatial anomaly*, this small region where the risk is larger than in the rest of the map. Therefore, a more appropriate name for this task should be *spatial disease anomaly detection* (**SDAD**).

SDAD is an unsupervised task where we have only a single instance of a disease map with the population counts and the number of observed cases in each area and our goal is to identify a hotspot subregion. However, the **SDAD** has some peculiarities, one of them being connected with the *spatial nature* of the dataset. Rather than many independent instances, as in the usual ML situation, the data take the form of a single map divided into small areas with the counts of disease cases in each of them, together with the underlying population sizes. In the disease map case, there is no possibility of carrying out the usual cross-validation based on training/test split of the disease map dataset. One may think of saving cases for a testing step by deleting areas from the map or deleting some cases from some areas. Both procedures would render either a highly biased or distorted dataset. In the first one, an incomplete map is used to scan for spatial anomalies in the complete map leading to inefficiency. In the second one, high risk spatial anomalies are underestimated. Hence, no training/testing split is undertaken, leading to overfitting and poor generalization capacity in the learning process.

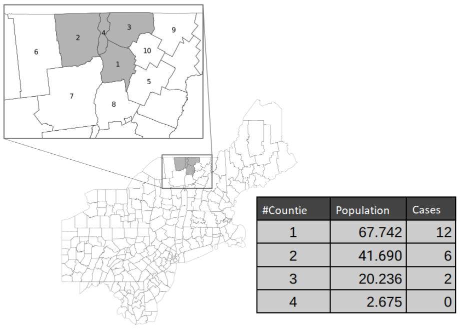
There is no repeated independent instantiating of this map to make up a usual i.i.d random sample as in ML tasks. As the true cluster and the underlying distribution are unknown, it is not trivial to generate other instances under the same model to evaluate the different spatial cluster detection methods. In this paper, we introduce model-free methods to generate several spatial maps from the same data-generating process as the single observed partitioned map. We propose two different methods, *a spatial bootstrap* and a *rewiring* method. We demonstrate that these random maps maintain the same spatial pattern as the observed data, not altering any unknown spatial signal the data may carry, such as the eventual presence of a true spatial disease cluster. The differences are only random disturbances not associated with any true underlying risk reflected in the original data. This provides a large number of datasets that leads to better generalization.

2 Definitions and Problem Statement

Let a map M be partitioned into $i = 1, \ldots, N$ areas. For each area i we have the underlying population n_i and the number of events y_i (e.g. disease cases) observed in a period of time. We assume that y_i is defined by a probabilistic distribution, such as $y_i \sim \text{Poisson}(\theta_i n_i)$ where θ_i is the incidence rate.

Figure 1 illustrates how the dataset is structured. It shows the counties in the Northeastern United States. For each county, we have the population at risk and the number of events observed. In this example, the population corresponds to the female population registered by the 1990 demographic census and the number of cases was synthetically generated and randomly distributed between the areas.

Our goal is to generate B perturbed versions of the map M where each area follows the same underlying distribution of M.



#Countie	Population	Cases
1	67.742	12
2	41.690	6
3	20.236	2
4	2.675	0

Fig. 1. Map with N small regions, each one with a certain number of events (e.g. disease cases) and the associated risk population. The table shows an example of data for the colored zones.

3 Sampling methods

We introduce two new methods to create new instances of a single disease map: the Bootstrap Samples and the Rewiring. The main idea in both methods is to create multiple rounds of cross-validation maps coming from the same statistical population and with the same spatial pattern present in the original dataset, *even if we do not know the true probability distribution that generates the observed data.*

Bootstrap samples. Let $Z_{ij} = 1$, if the j-th individual in area i is an event, such as a disease case, and $Z_{ij} = 0$, otherwise. With a disease rate (per capita) equal to θ, we have a binomial distribution for the number of cases:

$$(y_i | n_i) = \sum_{i=1}^{n_i} Z_{ij} \sim \text{Bin}(n_i, \theta) \approx \text{Poisson}(n_i \theta) \ .$$

The Poisson approximation is valid when θ is close to zero, which is usually the case for typical diseases such as site-specific cancers. Independently for each j-th individual in area i, generate the binary indicator $W_{ij} \sim \text{Bernoulli}(\pi)$ using a spatially constant probability hyper-parameter $\pi \in (0, 1)$. Then, conditionally on n_i, we have

$$(n_i^* | n_i) = \sum_{j=1}^{n_i} W_{ij} \sim \text{Bin}(n_i, \pi) \quad \text{and} \quad (y_i^* | n_i) = \sum_{j=1}^{n_i} W_{ij} * Z_{ij} \sim \text{Bin}(n_i, \pi\theta)$$

The collection $\{(n_i^*, y_i^*), i = 1, \ldots, N\}$ is a randomly thinned version of the original map with $\{(n_i, y_i), i = 1, \ldots, N\}$. The map with this π-thinned random sample of cases and population is called a *bootstrap sample* (or *map*).

The important point is that, if there is any spatial cluster in the original data, it is randomly reflected with the same spatial pattern in the bootstrapped map. The differences are due to random fluctuations that are not associated with the intrinsic disease rates θ_i. They are randomly thinned versions of the original map. Indeed, suppose that some areas have high-incidence disease rates producing a map with spatially varying rates. Let $y_i \sim \text{Bin}(n_i, \theta_i) \approx \text{Poisson}(n_i\theta_i)$. As the bootstrap probability π does not vary spatially, the observed map spatial pattern is reflected on the bootstrap map: $y_i^* \sim \text{Poisson}(n_i^* \pi \theta_i)$. Hence, the odds between disease rates from any pair of areas remain the same as in the original map. For example, while the odds between rates from areas i and j is θ_i/θ_j in the original map, it is equal to $(\pi\theta_i)/(\pi\theta_j)$ in the new bootstrap samples. The spatial pattern of the disease rates θ_i is retained in the bootstrapped map, whatever this spatial pattern is.

By randomly sampling W_{ij} independently, we generate B bootstrap maps with different disease and population counts out of our original data map. Each bootstrap disease map in this stack of B maps retains the same spatial pattern as the original map but differing in two respects. First, it has a smaller number of cases and population. Each area has approximately a proportion π of the original cases and population counts. Second, as the selection of retained cases and population is random, the bootstrapped maps differ between them. In short, this technique generates different maps from the real data but retaining the same spatial pattern, so one can test the resilience of machine learning algorithms with them.

Rewiring. Rewiring is a different way to generate pseudo maps that retains the spatial pattern from the original map but provides enough diversity to allow generalization capabilities for the algorithms. The population size is not altered, only the number of cases of each area can change. In contrast with the bootstrap approach, the total number of cases in the map, $\sum_i y_i$, is kept constant. The only change is that some of the observed cases may be randomly assigned to neighboring areas. The main idea is that each area may rewire a small proportion of its cases to neighboring areas. As all areas are rewiring, the expected number of cases in each area is kept approximately constant.

Let $\pi \in (0, 1)$ be a hyper-parameter. Rather than the truly observed number y_i of cases in each area, we generate a random number $y_i^* = K_i + R_i$ by keeping approximately a proportion π of its original number y_i (the K_i count) plus additional cases coming from the neighbors (the R_i count). More specifically, $K_i \sim \text{Bin}(y_i, \pi)$. The neighbors of area i receive the residual $y_i - K_i$ cases. This distribution is made according to a multinomial distribution, with the probability of selecting a given neighboring area proportional to its population. To be precise, let V_{ij} be a binary indicator with $V_{ij} = 1$ if areas i and j share boundaries, and $V_{ij} = 0$ otherwise. We set $V_{ii} = 0$. Then,

$$R_i = \sum_{j=1}^{N} V_{ji} * \text{Bin}\left(y_j - K_j, \frac{n_i}{\sum_k V_{jk} n_k} \right) .$$

As a consequence, the rewired expected disease rate in area i is given by

$$\begin{aligned}
\mathbb{E}\left(\frac{y_i^*}{n_i}\right) &= \pi\theta_i + \frac{1}{n_i}\sum_{j=1}^{N} V_{ji}\mathbb{E}\left((y_j - K_j)\frac{n_j}{\sum_k V_{ik} n_k}\right) \\
&\approx \pi\theta_i + \frac{1}{\nu n_i}\sum_{j=1}^{N} V_{ji}\mathbb{E}\left(y_j - K_j\right) = \pi\theta_i + \frac{1}{\nu n_i}\sum_j V_{ij} n_j \theta_j (1-\pi) \\
&= \pi\theta_i + \frac{1-\pi}{\nu}\sum_j V_{ij}\frac{n_j}{n_i}\theta_j \approx \pi\theta_i + \frac{1-\pi}{\nu}\nu\bar{\theta}_i \approx \theta_i
\end{aligned}$$

where $\bar{\theta}_i$ is the average values of the θ_j's that are neighbors of area i and ν is the average number of spatial neighbors of a given area in the map. For example, $\nu = 5.6$ for the USA continental counties.

Hence, the expected rewired rate in each area is the same as in the original unknown mechanism that generates the observed data. As in the bootstrapped maps, the rewired maps will retain approximately whatever spatial pattern is present in the original map. This technique simulates scenarios where an incorrect assignment of cases to regions is possible, so one can test the resilience of the algorithms under this circumstance.

4 Discussion

The choice of the π parameter is important in both, the Rewiring and Bootstrap methods. With π equals to 1, we have no variation between the instances and all generated instances will be exactly the same as the original data. On the other hand, the smaller the value of π, the greater the randomness added and, therefore, more distant from the spatial pattern of the original data the perturbed instances will be. Ideally, the value of π should be close to but not equals to 1.

Both Bootstrap Samples and Rewiring methods achieve the same goal: generating perturbed versions of a partitioned map maintaining the underlying distribution of the data. However, there are some particularities that should be taken into account when choosing one of them. The choice depends on the task and the sparsity of the data.

The Rewiring method can occasionally assign some events to an area where no event was observed in the original map. Therefore, if it is important to keep these areas with no events, the Bootstrap Samples method should be the better choice. However, if the data is sparse, with just a few events observed in most areas, Rewiring could be preferable, since Bootstrap Samples can increase the sparsity.

5 Conclusion

In this work we propose two new methods to augment spatial datasets: Bootstrap Samples and Rewiring. To the best of our knowledge our methods are the first ones that can generate many perturbed versions from a single map where the data is aggregated by geographic sub-regions. Furthermore, they do not need any prior information or assumption about the dataset. We demonstrated that the generated perturbed data follow the same distribution of the original map and preserve important characteristics, such as any hotspot that may be present or correlation between adjacent areas. As future work, we suggest studies about the impact on the choice of hyperparameter π and approaches to determine its value deterministically.

References

1. Abrams, B., Anderson, H., Blackmore, C., Bove, F.J., Condon, S.K., Eheman, C.R., Fagliano, J., Haynes, L.B., Lewis, L.S., Major, J., et al.: Investigating suspected cancer clusters and responding to community concerns: guidelines from cdc and the council of state and territorial epidemiologists. Morbidity and Mortality Weekly Report: Recommendations and Reports **62**(8), 1–24 (2013)
2. Ciregan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3642–3649. IEEE (2012)
3. Duong, H.T., Nguyen-Thi, T.A.: A review: preprocessing techniques and data augmentation for sentiment analysis. Computational Social Networks **8**(1), 1–16 (2021)
4. Ghaffar, M., McKinstry, A., Maul, T., Vu, T.: Data augmentation approaches for satellite image super-resolution. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences **4** (2019)
5. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25**, 1097–1105 (2012)
6. Kulldorff, M.: A spatial scan statistic. Communications in Statistics-Theory and methods **26**(6), 1481–1496 (1997)
7. Nalepa, J., Marcinkiewicz, M., Kawulok, M.: Data augmentation for brain-tumor segmentation: a review. Frontiers in computational neuroscience **13**, 83 (2019)
8. Nogueira, K., Penatti, O.A., Dos Santos, J.A.: Towards better exploiting convolutional neural networks for remote sensing scene classification. Pattern Recognition **61**, 539–556 (2017)