Lecture 16 — May 28

Lecturer: Lester Mackey *Scribe: Ankur Bapna*

16.1 Learning with Missing Data

Suppose that we observe a dataset where some of the features x_{ij} are missing. Such data may arise from non-response in surveys, censorship or early dropout in longitudinal trials, corrupted results or measurements, different features collected from different studies, or recommender systems, where each user only explicitly expresses his or her preferences for a subset of available items. In this lecture and the next, we will learn how to carry out unsupervised learning in the presence of missing data.

16.1.1 Types of missingness

Since not all missing data is the same, we begin by considering several important classes of missingness. We say that a feature is **missing completely at random (MCAR)** if the probability that the feature is missing is independent of the value of the feature and the values of any other features. This is often the best-case missingness scenario. This would occur, for example, if only a uniformly random subset of the respondents to our survey answer question 3.

We say that a feature is **missing at random (MAR)** if the probability that a feature is missing is independent of the value of the feature. In this case, the probability of being missing can depend on the value of the other features. For example, this would occur if a survey respondent who answers "yes" to "Are you forgetful?" is more likely to forget to answer question 3.

Finally, we say that a feature is **not missing at random** if the probability that a feature is missing can depend on the value of the feature. This would arise if a respondent did not answer question 3 because the answer was embarassing. This scenario is typically the most difficult to deal with.

16.1.2 Testing for MCAR: Supervised Learning

Supervised learning methods can be used to test the assumption of MCAR in categorical covariates in the following way:

1. Code 'missing' as a new category.
2. Run a supervised analysis (to predict a separate target variable) and check if 'missing' has an effect on the prediction of the response in the learned model.
3. If the category 'missing' has an effect, this is evidence that data is not MCAR.

Note that this test could be ported to the unsupervised learning setting by treating one of the features as the target variable and predicting it using the remaining variables.

16.2 Dealing with missing data

We next consider various strategies for learning with missing data.

16.2.1 Datapoint or feature deletion

An especially convenient option is to delete datapoints with missing features or to delete features with missing values and then apply standard learning algorithms to resulting dataset. However, this can be problematic if missing data is common, the dataset is small, or all features are valuable for our analysis. In these cases, we typically resort to other methods.

16.2.2 Imputing missing features (single imputation)

Imputing missing features has the benefit of enabling us to apply standard learning algorithms after the imputation. However, the resulting inferences may be strongly influenced or biased by the imputation choice, particularly when the fraction of missing data in our dataset increases. Let us consider a few standard approaches to imputing missing features.

A common, simple, and inexpensive strategy is to **impute the mean or median observed value of the feature**. A downside is that this strategy completely ignores the correlation among the features which could be used to find better estimates for the missing feature. **Imputation by regression on the other features** attempts to capture the relationships between various features. However, it also has several drawbacks: it is difficult to apply unless (a) we observe the same features for many datapoints or (b) the regression method used can handle missing data. Also, (c) this imputation scheme only accounts for the part of the feature that can be predicted from other features.

To deal with (a), we can perform **iterated regression** where we first impute all the missing values using some baseline technique (like imputing the means) and then iteratively refine imputations by regressing each feature with missing values on the remaining features (making use of the imputations from the prior round for the regression covariates).

With regard to (b), we several supervised learning procedures can cope with missing data out of the box. For tree based methods like CART **treat 'missing' as a special value**. We can also modify methods to deal with missing values. For example, in k-nearest neighbors and other methods based on pairwise dissimilarity, we can modify a pairwise dissimilarity measure (like mean squared distance) to be computed only over those features which are observed for both datapoints.

To address point (c), we could perform **multiple imputation** which better accounts for the uncertainty inherent in imputation by forming a posterior distribution over unobserved values and sampling many possible imputations from this distribution. Each imputed dataset can then be used for follow-up analysis.

Yet another imputation strategy that arises in the context of time series or other sequential data, where a strong relationship exists among neighboring values, is to impute by

interpolating between neighboring values.

16.2.3 Adapting unsupervised learning procedures

A different approach to coping with missing data is to modify our unsupervised learning techniques to deal with missing data directly. We will adopt the following general strategy for performing this adaptation:

1. Frame the unsupervised learning problem as a data reconstruction problem.
2. Modify the objective to reconstruct only the observed features.

Notably, once our unsupervised representations are learned in this fashion, they can also be used to perfom imputation (as a added bonus).

16.3 PCA with Missing Data

As an example, we will apply this adaptation strategy to PCA. Recall the PCA reconstruction objective,

$$\begin{array}{ll} \underset{U \in \mathbb{R}^{p \times k}}{\text{minimize}} & \Sigma_{i=1}^{n} \|x_i - UU^T x_i\|_2^2 \\ \text{subject to} & U^T U = I. \end{array}$$

This problem is equivalent to

$$\begin{array}{ll} \underset{U \in \mathbb{R}^{p \times k},\, z_i \in \mathbb{R}^k}{\text{minimize}} & \Sigma_{i=1}^{n} \|x_i - Uz_i\|_2^2 \\ \text{subject to} & U^T U = I, \end{array}$$

since, for any U, the optimal value of z_i is $U^\top x_i$. This problem is also equivalent to

$$\begin{array}{ll} \underset{M \in \mathbb{R}^{n \times p}}{\text{minimize}} & \|X - M\|_F^2 \\ \text{subject to} & \text{rank}(M) \leq k \end{array}$$

where $X \in \mathbb{R}^{n \times p}$ is the data matrix with the i-th row equal to x_i^T. This follows from the fact that an optimal solution is given by the rank-k truncated SVD of X, $M^* = V \Sigma_k U^T$. Therefore, PCA recovers the best rank-k reconstruction of the data matrix X.

Now let Ω be the set of all observed features, so that x_{ij} is observed iff $(i, j) \in \Omega$. Then our missing data objective for the PCA problem is

$$\begin{array}{ll} \underset{M \in \mathbb{R}^{n \times p}}{\text{minimize}} & \underset{i,j \in \Omega}{\Sigma} \|x_{ij} - m_{ij}\|_2^2 \\ \text{subject to} & \text{rank}(()M) \leq k, \end{array}$$

where m_{ij} represents an entry of M. Unfortunately, this problem is non-convex (like the original PCA problem) and has no closed form solution in general (unlike the original PCA problem). Hence, a variety of strategies are available for finding approximate solutions.

16.3.1 Iterated SVD

A popular solution for relatively small data matrices is sometimes termed the **iterated SVD**. To develop this optimization scheme, we first note that an equivalent formulation for our missing data PCA problem is

$$\begin{aligned} \underset{M \in \mathbb{R}^{n \times p}, \tilde{X} \in \mathbb{R}^{n \times p}}{\text{minimize}} \quad & \|\tilde{X} - M\|_F^2 \\ \text{subject to} \quad & \text{rank}(M) \leq k \\ & \tilde{x}_{ij} = x_{ij}, \, \forall (i,j) \in \Omega \end{aligned}$$

Now we can perform block coordinate descent (alternating minimization) on the two optimization variables $\tilde{X}$ and M. That is, we repeat the following two steps until convergence is achieved.

1. **Imputation:** Update $\tilde{X}$ given M using $\tilde{x}_{ij} = m_{ij} \, \forall (i,j) \in \Omega$.
2. **Truncated SVD:** Update M for fixed $\tilde{X}$ using $M =$ rank-k truncation of the SVD of $\tilde{X}$.

One can also view this approach as an alternating projection algorithm for finding a point M in the intersection of two sets, the set of rank-k matrices and the set of matrices $\{N : n_{ij} = x_{ij}, \, \forall (i,j) \in \Omega, \, N \in \mathbb{R}^{n \times p}\}$. While this is a viable approach for smaller matrices, it is prohibitively expensive for large matrices with large amounts of missingness. For example, most recommender systems take advantage of the substantial sparsity in their observation matrices (in the Netflix Prize dataset only 1% of all possible preferences were observed), so simply imputing and storing all np possible entries may be prohibitive. Moreover, since our objective is non-convex, this method is subject to suboptimal stationary points. Next time, we will explore more practical and scalable strategies to solving the missing data PCA problem that avoid explicit imputation.