Ensemble Classification for Queries with Missing Values

Mathieu Charbonnel, Robert J. Durrant
Waikato University

Abstract

We consider the problem of classification when queries (i.e. vectors of predictors to be classified) may have missing values. Unlike previous work on imputation, which focuses on missingness in training examples, here we consider the fact that the mechanism for missingness in queries may not be the same as that which causes missing values in the training set. Such queries can arise in practice when, for example, values may be missing either due to chance or due to the actions of an adversary. Therefore such queries may have values which are missing at random, missing completely at random, or missing not at random.
In this work we take a very simple data-driven approach towards solving this problem; namely, we use a generative classifier to compare the likelihood of several versions of the query – each version with missing values imputed according to a different model of missingness – in order to estimate its class.
As a concrete example of this approach we present experiments using a simple variant of Fisher's linear discriminant and we show that its accuracy on synthetic and real-world datasets with varying proportions of missing data, under different missingness mechanisms, is improved using our approach.

1 Introduction

Missing data are ubiquitous in real world data analysis tasks and, since the mechanism leading to missingness is frequently unknown, present a serious challenge to predictive modelling. For example, values could be missing due to a broken sensor or communication pipeline, or missing in the responses to a form or questionnaire, and these situations could arise due to chance, due to the state of other values we can measure, or due to the actions of an adversary or unwilling respondent. In particular values may be 'missing completely at random' (MCAR), in which case the absence of a value depends on neither observed nor unobserved values, 'missing at random' (MAR) in which case absence can depend only on observed values, and 'missing not at random' (MNAR) in which case absence depends on *unobserved* values. The MCAR and MAR cases are often referred to as 'ignorable' cases, since the missing values can be estimated from the sample when modelling the data generator. On the other hand when values are MNAR one cannot rely on sample information by itself, since the presence or absence of a variable may depend on its own unobserved value, for example. Various approaches have been proposed to manage the problem effectively, especially from the perspective of modelling. The most common approaches for dealing with missing values, when modelling the data generator, include 'complete case' analysis (i.e. discarding any observations with missing values), imputing missing values with the corresponding value obtained from another observation (e.g. 'hot-' and 'cold-deck' imputation, and nearest neighbour imputation), imputing a point estimate of any missing values (e.g. replacing the missing values with the sample mean (say) of the observed values), or multiple imputation of missing values (e.g. replacing missing values several times with values sampled from an appropriate distribution). Under different models of missingness these approaches may furnish biased or unbiased estimates of the missing values and, for values MNAR, all of these approaches will generally furnish biased estimates of the data generator.
However we can observe that, for prediction, one is not mainly interested in modelling the data generator faithfully *per se* but rather the goal is to maximise the predictive accuracy of the model, and this may not require unbiased estimates of the data generator – parameter estimation for the generative model is simply one means to this end. Indeed when values are MNAR

but the absence of a value is believed to be predictive of the class then missingness can be treated as a factor, for example, which can aid prediction despite the fact that unbiased estimation of the data generator may not be possible.

On the other hand the missingness mechanism may vary for different training examples with the same missing variables, or it may be different for training observations than for queries (unlabelled observations). In particular, the missingness mechanism in training data (where the label is observed) could be MAR given the class labels, but for a query (where the label is unobserved) the same mechanism is effectively MNAR.

In this work we propose a simple data-driven approach for the classification of queries with missing values using a generative classifier, by comparing the model likelihood of the class label given the query under three different models for missingness. In particular, our approach is to impute the missing values in three different ways which correspond to values MCAR, MAR, and MNAR under the simplifying assumption that – if values are MNAR – missingness in the query is dependent only on the unobserved class label.

Our approach is very general, and in principle could be applied to any generative classifier *mutatis mutandis*, but here for concreteness we explore a simple variant of Fisher's Linear Discriminant (FLD) in our experiments. Note that we do not claim FLD is an optimal, or even a good, choice of classifier for the datasets we study. Instead our focus here is on exploring (1) whether the added complexity of our approach improves prediction, (2) the effect of different commonly-used imputation procedures on the accuracy of our procedure, and (3) if the 'best' imputation method for a particular missingness mechanism be learned from training data.

2 Problem Statement

We consider the statistical learning problem of two-class classification with generative (parametric) models. Under this approach the famous Neyman-Pearson lemma states that, for classes $\{0, 1\}$ with corresponding densities specified by parameters θ_0, θ_1 respectively, the uniformly most powerful test for class membership of a query point x is given by the likelihood ratio test:

$$h(x) := \mathbf{1}\left(\frac{f(x|\theta_1)}{f(x|\theta_0)} > 1\right)$$

where $f(x|\theta)$ is the likelihood of x given that it belongs to the density with parameters θ and $\mathbf{1}(\cdot)$ is the indicator function which returns 1 if its argument is true and 0 otherwise. Since the parameters $\theta := (\theta_0, \theta_1)$ are unknown they are estimated by $\hat{\theta}$ giving the practical decision rule:

$$\hat{h}(x) := \mathbf{1}\left(\frac{f(x|\hat{\theta_1})}{f(x|\hat{\theta_0})} > 1\right) \tag{1}$$

and, in light of the Neyman-Pearson lemma it is usual to choose an estimator $\hat{\theta}$ that is UMVUE or has maximum likelihood for the given parametric family.

Now consider the case where values are missing from some of the variables in a query point x – in this situation the likelihood ratio in (1) is undefined and one cannot even begin to make a meaningful estimate of the class label without 'fixing' the query in some way[1]. In particular, for a query one cannot simply discard the query with missing values. Plugging in an arbitrary value such as 0 for the missing value(s) or setting the values as missing (e.g. using 'NA' in R) fixes this problem in some sense but, as we shall see later, at the cost of a great deal of accuracy in the estimated labels. Therefore one should carry out some form of principled (i.e. theoretically-motivated) imputation of the missing values, and we should then evaluate:

$$\hat{h}(x) := \mathbf{1}\left(\frac{f(x^I|\hat{\theta_1})}{f(x^I|\hat{\theta_0})} > 1\right) \tag{2}$$

where in (3) x^I is the query with missing values imputed in such a principled way. Unfortunately this does not completely solve the problem, since the imputation method one should theoretically use depends on the mechanism for missingness (MCAR, MAR or MNAR) and this may not be known for the query.

3 Our proposed approach

In this work we propose a simple approach for dealing with this latter problem – we will impute values in (3) according to different implied missingness mechanisms and use the scores obtained from the likelihood ratios in (3) to decide our estimate of the class label. Thus our scheme can be viewed as a simple classifier ensemble approach where, instead of passing a query to several different classifiers in order to reach a decision we instead pass several different (imputed) versions of the same query to the same classifier in order to reach a decision. To the best of our knowledge this approach has not been considered before.

For concreteness and simplicity, we will focus on the case where the class conditional distributions are modelled as Gaussians with different means but identical covariance matrices. This leads to the following model,

[1] One could, of course, simply assign the most common label in these cases but this amounts to throwing away all of the information still contained in the observed variables in the query.

which is called Fisher's linear discriminant:

$$\hat{h}_{FLD}(x) := \mathbf{1}\left(\frac{2\pi^{-p/2}|\hat{\Sigma}^{-1}|}{f(x^I|\hat{\theta_0})} > 1\right) \quad (3)$$

Usually we want to compare the likelihood of x belonging to class 0, $(x-\mu_0)^t \sum^{-1}(x-\mu_0)$ with thelikelihood of x belonging to class 1, $(x-\mu_1)^t \sum^{-1}(x-\mu_1)$, adding a constant if the two classes are not balanced. This is designed to work with a multivariate normal distribution and provides the Bayes optimal solution in that case (more on that later). In more general cases it can still be interpreted as computing how scattered the elements from a given class are compared to the scatter between elements from the two different classes. Imputation before LDA is usually done that way for a given feature: computing the mean of all the values that are not missing for that feature and plugging this mean each time a value for that feature is missing. A logical way for a more realistic imputation is to consider before the training to which class each new element belongs to plug a mean corresponding to the right class. To achieve that, we just need for each feature that is likely to have missing value to ncompute the mean for each class. Then, the principle when testing is, instead of imputing the grand mean in x's missing coefficients before deciding (using this decision function:$(x-\mu_0)^t \sum^{-1}(x-\mu_0)-(x-\mu_1)^t \sum^{-1}(x-\mu_1))$, to impute class 0 means for the likelihood to belong to class 0 and to impute class 1 means for the likelihood to belong to class 1 : $(x_0-\mu_0)^t \sum^{-1}(x_0-\mu_0)-(x_1-\mu_1)^t \sum^{-1}(x_1-\mu_1)$. The change brought by this new decision function is subtle compared to just taking the grand mean. About the training set it's mainly about reaching a good accuracy with smaller datasets. But about the decision function, as imputing with class 0 or class 1 increase both $(x_0-\mu_0)^t \sum^{-1}(x_0-\mu_0)$ and $(x_1-\mu_1)^t \sum^{-1}(x_1-\mu_1)$ the question is to determine how those changes compete together.

4 Note on the meaning of Fisher Discriminant Analysis on multivariate normal law

The multivariate Gaussian distribution, or joint normal distribution has several equivalent definitions which ensure the variables have good properties when using Fischer Discriminant Analysis. One definition is that a random vector is said to be k-variate normally distributed if every linear combination of its k components has a univariate normal distribution. From this hypothesis it is also possible to show that density of class 0 variables is given by the formula $(x-\mu_0)^t \sum^{-1}(x-\mu_0)$. We can see that in that situation LDA just compares the density function of the given x for each class and decides in favour of the class that shows the biggest density at that point. The space is separated into two subspaces depending on which density of probability is bigger.

5 Generation of incomplete data

This method (conditional Imputation) as well as many classic others were tried on real life datasets to be compared. But we also generated multivariate normal distributions and removed variables completely at random, at random or not at random. The code, which is on the GitHub (https://github.com/mathieu-charbonnel/ImputationBeforeFisherDiscriminant Analysis.git) gives the possibility to change a lot of different parameters and compare the accuracy of this very LDA classification model after having imputed the data with all the available methods. We chose to generate the variables for the two different classes with different mean vectors (obviously) but the same covariance matrix to make the use of LDA lighter and ease comparisons.We worked with 4 different types of covariance matrices: most commonly the one we call "random" is simply generated with the python sklearn function $make_spd_matrix(dim, random_state = None)$ to generate random symetric positive-definite matrix, "decreased correlation" is as follows: cov[i][i]=1 and cov[i][j]=exp(abs(i-j)) if i and j are different. Two variants were coded but not used in this article: 'strong correlation with higherIndex' where cov[i][i]=i and cov[i][j]=exp(abs(i-j))*max(i,j) if i and j are different, and finaly purely diagonal matrices. We also normalized the distance between the mean vectors to obtain more consistent results across the different generated datasets : to do so we used the Kullback-Leibler divergence which has a very simple form when comparing two multivariate normal distributions of the same dimension: $D(N_1|N_2) = \frac{1}{2}(Tr(\sum_1^{-1}\sum_0) + (\mu_1-\mu_0)^t \sum^{-1}(\mu_-\mu_0) - k + ln(\frac{det(\sum_1)}{det(\sum_0)}$, translating into $(\mu_1-\mu_0)^t \sum^{-1}(\mu_1-\mu_0)$when $\sum_0 = \sum_1$.
Choices had to be made about how to remove data depending on the type of missingness we wanted to simulate. Removing values completely at random with a given probability is relatively straightforward. For data missing at random we generated random coefficients that we multiplied with the variables other than the potentially missing one. If the sum was more than a given parameter (chosen so that the overall probability of missingness is what we want), then the value was removed. That way missingness depends on observable variables. For data missing not at random, we could either remove one value if it was over or under a well- chosen threshold or simply randomly removing data from only one class.

We chose the second option as the simplest and easiest to exploit for classification. At the end we could have a lot of parameters vary to compare the efficiency of different imputation methods depending on these, type of missingness, the overall probability of missingness, size of the training set, dimension of the problem, covariance matrix (very correlated variables or not). We worked with 4 different types of covariance matrices

6 Imputation experiments on synthetic data

Covariance matrix is random symmetric positive semi definite
When probability of missing data is 0.1:

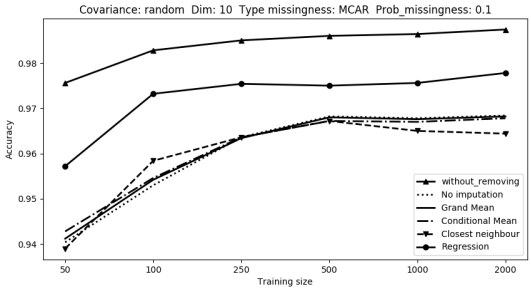

Figure 1: MCAR

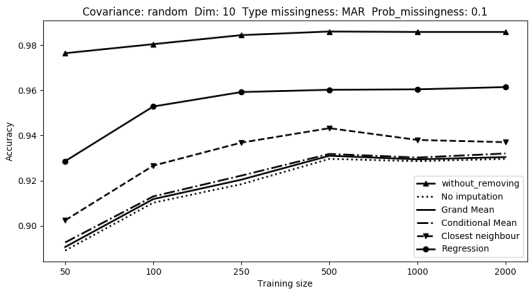

Figure 2: MAR

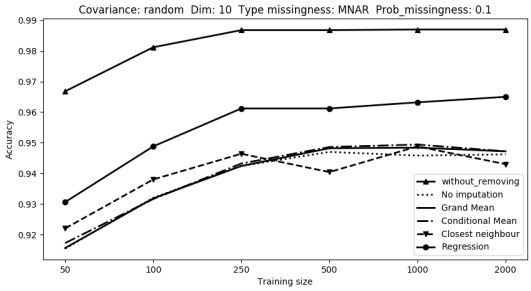

Figure 3: MNAR

As we can see using regression every time to make good use of the correlation between the features to predict the missing ones seems to be the best option when missingness rate is relatively low. Let us note that for little training sets nearest neighbour can be considered.
For half of the data missing:

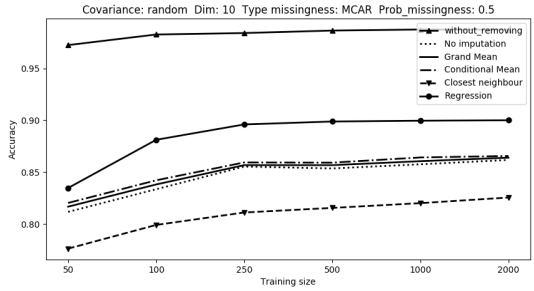

Figure 4: MCAR

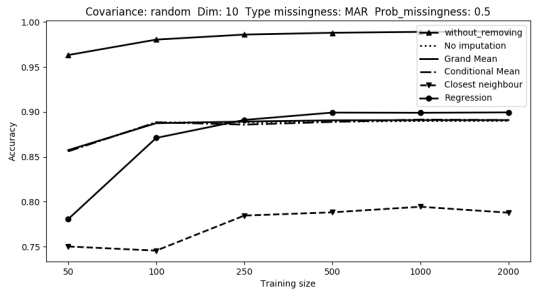

Figure 5: MAR

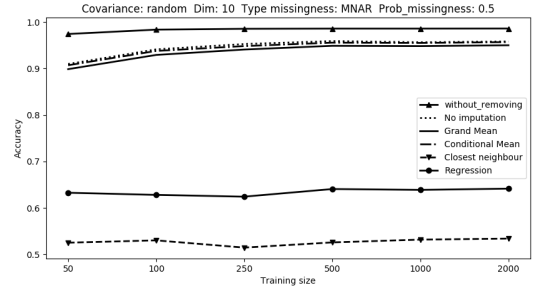

Figure 6: MNAR

With more missing data, Regression stays interesting when data is missing completely at random but when missingness tells something about either the other features (MAR) or the class (MNAR) using this information with methods like conditional mean or not imputing at all can prove better.

During the tests we wondered how scattered the accu-

racy of the different models were across the randomly generated datasets. Here is the answer.

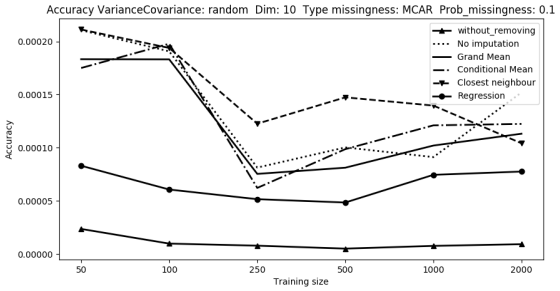

Figure 7: Accuracy Variance

Introduction of multiple imputation:

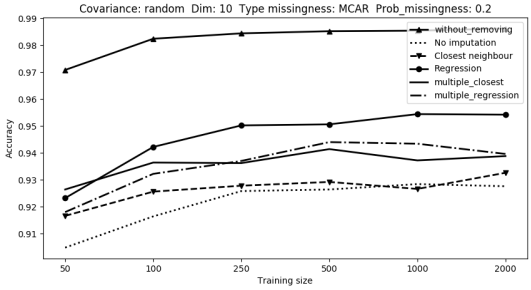

Figure 8: Multiple imputation with missingness of 0.2

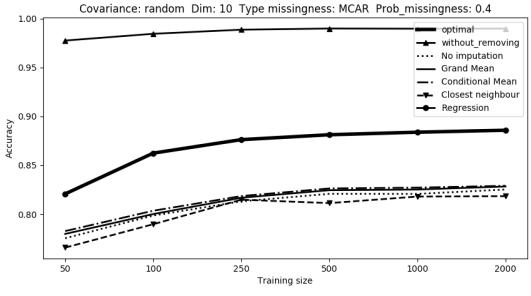

Figure 9: Multiple imputation with missingness of 0.4

Majority vote after imputing with several points along the line between the two nearest neighbours, and majority vote after imputing with points normally distributed around regression estimate. While multiple imputation enhances nearest neighbour imputation a lot it seems that in our case of pre generated data (so following perfect normal distribution) it is better to stick to deterministic regression.

Testing the optimal imputation:

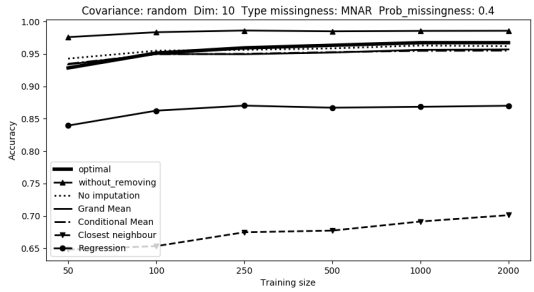

Figure 10: Optimal imputation

The optimal imputation is chosen with training set and applied it on testing set each time (and we take the average) This new flexibility could have played in our favour or not as it is only based on accuracy levels on training set. Overall this flexibility is only useful in MNAR case. Indeed, we saw that conditional mean and regression could both be best in MNAR case.

Test with different covariance matrices
Here we test our results on datasets generated with another covariance matrix which reduces correlation between the features: basically 1 on the diagonal and exp(-abs(i-j)) the rest of the time. Similar results with overall diminished accuracy for regression and nearest neighbours which both make use of correlation.

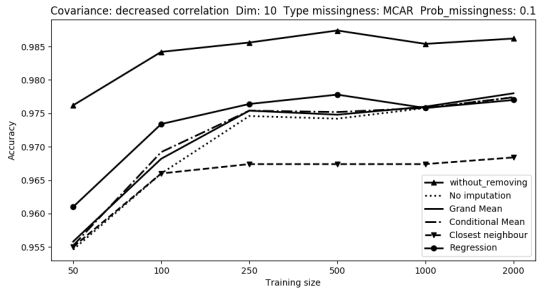

Figure 11: reduced correlation

Test with another classification method : naive bayes. The accuracy is overall far worse than with lda. It shows similar trends as with lda but also shows an interesting appearance of closest neighbour as a good imputation method when the distribution does not exactly fit the classification method (like multivariate and lda did)

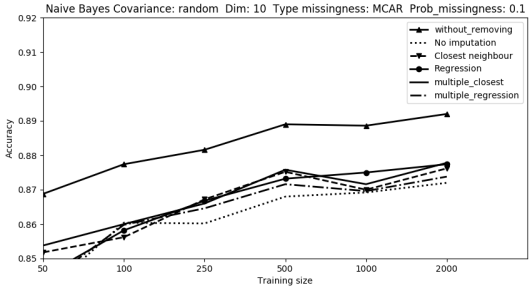

Figure 12: naive bayes

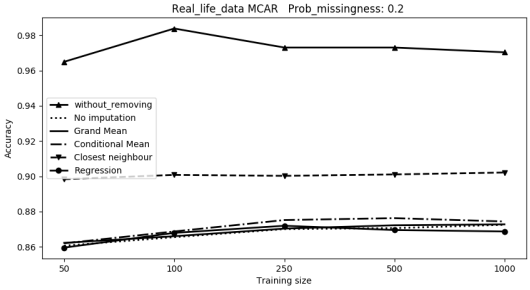

Figure 14: Missingness completely at random

7 Imputation experiments on real data

The dataset we chose to be using is a public banknote authentification dataset. It can be found at the following URL: https://archive.ics.uci.edu/ml/datasets/banknote+authentication. Four features are obtained from 400 x 400 pixels images by using a Wavelet transform tool.

- Variance of Wavelet Transformed image
- Skewness of Wavelet Transformed image
- Curtosis of Wavelet Transformed image
- Entropy of image
- Class

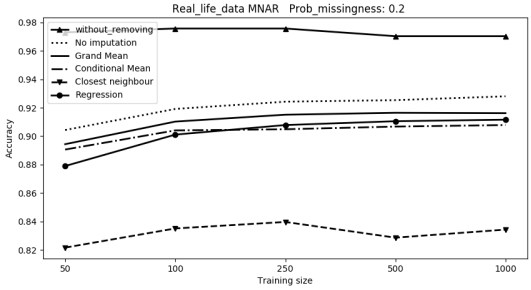

Figure 13: Missingness not at random

When data does not follow clear normal distribution, nearest neighbours starts being competitive again when data is missing completely at random!

We had already seen that imputation was overall not efficient when data is missing not at random, it got verified here with no imputation on top.

8 Conclusion

Overall the trends we spotted across the different datasets/ classification methods that we used is that with clean normal distribution and correlated variables, regression is on top. However with more noisy data sets or classification methods not very fit for the distributions, multiple imputation with closest neighbour could actually perform better. When data is missing not at random with high degree of missingness not imputing can be the reasonable choice.
When using the Linear Discriminant Analysis method, the original approach from this work that we called conditional mean revealed to perform better overall than grand mean so it may be used instead in similar cases.
If the type/probability of missingness is unknown, we saw that testing the imputation methods on the training set to chose the best one in the given situation is a safe and reasonable idea, and leads to very good results.