



## Classification on Missing Data for Multiple Imputations

**A. Nithya Rani**

M.C.A., M.Phil., M.B.A., Assistant Professor, Dept of  
Computer Science, C.M.S College of Science and  
Commerce, Coimbatore, Tamil Nadu, India

**Dr. Antony Selvdoss Davamani**

Reader in Computer Science, NGM College  
(Autonomous), Pollachi, Coimbatore,  
Tamil Nadu, India

### ABSTRACT

This research paper explores a variety of strategies for performing classification with missing feature values. The classification setting is particularly affected by the presence of missing feature values since most discriminative learning approaches including logistic regression, support vector machines, and neural networks have no natural ability to deal with missing input features. Our main interest is in classification methods that can both learn from data cases with missing features, and make predictions for data cases with missing features.

**Keywords:** [Multiple imputations, Classification, Machine learning approaches]

### 1. INTRODUCTION

We begin with an overview of strategies for dealing with missing data in classification. Generative classifiers learn a joint model of labels and features. Generative classifier does have a natural ability to learn from incomplete data cases and to make predictions when features are missing. We then discuss several strategies that can be applied to any discriminative classifier including case deletion, imputation, and classification in subspaces. Finally, we discuss a frame-work for classification from incomplete data based on augmenting the input representation of complete data classifiers with a vector of response indicators.

We consider Linear Discriminant Analysis as an example of a generative classifier. We present both maximum likelihood and maximum conditional

likelihood learning methods for a regularized Linear Discriminant Analysis model with missing data. We consider applying these methods to classification with missing data using imputation, reduced models, and the response indicator framework.

### 2. Frameworks for Classification with Missing Features

Generative classifiers have a natural ability to deal with missing data through marginalization. This makes them well suited for dealing with random missing data. The most well-known methods for dealing with missing data in discriminative classifiers are case deletion, imputation, and learning in subspaces. All of these methods can be applied in conjunction with any classifier that operates on complete data. In this section we discuss these methods for dealing with missing data. We also discuss a different strategy for converting a complete data classifier into a classifier that can operate on incomplete data cases by augmenting the input representation with response indicators.

#### 2.1 Generative Classifiers

Generative classifiers model the joint distribution of labels and features. If any feature values are missing they can be marginalized over when classifying data cases. In class conditional models like the Naive Bayes classifier and Linear Discriminant Analysis, the marginalization operation can be performed efficiently. Missing data must also be dealt with during learning. This typically requires an application of the Expectation Maximization algorithm. However,

generative classifiers require making explicit assumptions about the feature space distribution, while discriminative classifiers do not.

## 2.2 Classification and Imputation

Imputation is a strategy for dealing with missing data that is widely used in the statistical community. In unconditional mean imputation, the mean of feature  $d$  is computed using the data cases where feature  $d$  is observed. The mean value for feature  $d$  is then used as the value for feature  $d$  in data cases where feature  $d$  is not observed. In regression imputation, a set of regression models of missing features given observed features is learned. Missing features are filled in using predicted values from the learned regression model.

Regression and mean imputation belong to the class of single imputation methods. In both cases a single completion of the data set is formed by imputing exactly one value for each unobserved variable. Multiple imputations is an alternative to single imputation procedures. As the name implies, multiple completions of a data set are formed by imputing several values for each missing variable. In its most basic form, the imputed values are sampled from a simplified imputation model and standard methods are used on each complete data set. The principal advantage of multiple imputations over single imputation is that multiple imputation better reflects the variability due to missing values. Sophisticated forms of multiple imputations are closely related to approximate Bayesian techniques like Markov chain Monte Carlo methods, and can be viewed as an approximation to integrating out the missing data with respect to an auxiliary distribution over the feature space.

The key to imputation techniques is selecting an appropriate model of the input space to sample from. This is rarely the case in single imputation where imputing zeros is common. A standard practice in multiple imputations is to fit a Gaussian distribution to each class, and sample multiple completions of the missing features conditioned on the observed features. More flexible imputation models for real valued data are often based on mixtures of Gaussians. In high dimensions, learning a mixture of probabilistic principal components analysis or factor analysis models may be more appropriate.

The advantage of imputation methods is that they can be used in conjunction with any complete data classifier. The main disadvantage is that learning one or more imputation models can be a costly operation. In addition, using multiple imputations leads to maintaining an ensemble of classifiers at test time. Combining multiple imputations with cross validation requires training and evaluating many individual classifiers.

## 2.3 Classification in Sub-spaces: Reduced Models

Perhaps the most straightforward method for dealing with missing data is to learn a different classifier for each pattern of observed values. Sharpe and Solly studied the diagnosis of thyroid disease with neural networks under this framework, which they refer to as the network reduction approach. The advantage of this approach is that standard discriminative learning methods can be applied to learn each model. Sharpe and Solly found that learning one neural network classifier for each subspace of observed features led to better classification performance than using neural network regression imputation combined with a single neural network classifier taking all features as inputs.

As Tresp et al. point out; the main drawback of the reduced model approach is that the number of different patterns of missing features is exponential in the number of features. In Sharpe and Solly's case, the data set contained four inputs, and only four different patterns of missing features, making the entire approach feasible.

## 2.4 A Framework for Classification with Response Indicators

An alternative to imputation and subspace classification is to augment the input to a standard classifier with a vector of response indicators. The input representation  $\mathbf{x}e_n = [\mathbf{x}_n \mathbf{r}_n; \mathbf{r}_n]$  can be thought of as an encoding for  $\mathbf{x}^o_n$ . Here signifies elementwise multiplication. A trained classifier can be thought of as computing a decision function of the form  $f(\mathbf{x}^o_n)$ . In logistic regression, multi-layer neural networks, and some kernel-based classifiers, substituting  $\mathbf{x}e_n$  for  $\mathbf{x}_n$  is the only modification required. This framework was studied in conjunction with certain SVM models, although they focus on the problem of structurally incomplete data cases. Structurally incomplete data cases arise when certain feature values are undefined for some data cases.

### 3. Linear Discriminant Analysis

In this section we present Linear Discriminant Analysis, and its application to classification with missing features. We begin by reviewing Fisher's original conception of Linear Discriminant Analysis. We then describe the relationship between Fisher's view and a view based on maximum probability classification in a class conditional Gaussian model. We discuss several extensions of LDA including Quadratic Discriminant Analysis (QDA), and Regularized Discriminant Analysis (RDA). We introduce a new method for missing data classification based on generative training of a linear discriminant analysis model with a factor analysis-

style co-variance matrix. Finally, we present a discriminative training method for the same model that maximizes the conditional probability of labels given features.

#### 6.2.5 LDA and Missing Data

As a generative model, Linear Discriminant Analysis has a natural ability to deal with missing input features. The class conditional probability of a data vector with missing input features is given in Equation 1. The posterior probability of each class given a data case with missing features is shown in Equation 2

$$P(\mathbf{X}_n^o = \mathbf{x}_n^o | Y_n = c) = |2\pi\Sigma^{oo}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x}_n^o - \mu_c^o)^T (\Sigma^{oo})^{-1} (\mathbf{x}_n^o - \mu_c^o)\right) \quad \text{---- (1)}$$

$$P(Y = c | \mathbf{X}_n^o = \mathbf{x}_n^o) = \frac{\theta_c |2\pi\Sigma^{oo}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x}_n^o - \mu_c^o)^T (\Sigma^{oo})^{-1} (\mathbf{x}_n^o - \mu_c^o)\right)}{\sum_c \theta_c |2\pi\Sigma^{oo}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x}_n^o - \mu_c^o)^T (\Sigma^{oo})^{-1} (\mathbf{x}_n^o - \mu_c^o)\right)} \quad \text{---- (2)}$$

#### Maximum Likelihood Estimation

The maximum likelihood estimate of the mean parameters is computed from incomplete data as shown in Equation 3

$$\mu_{dc} = \frac{\sum_{n=1}^N [y_n = c][r_{dn} = 1]x_{dn}}{\sum_{n=1}^N [y_n = c][r_{dn} = 1]}$$

The parameters of the full covariance matrix can be estimated using the Expectation Maximization algorithm. However, when data vectors are high dimensional and there are a relatively small number of data cases, it is preferable to use a structured covariance approximation. We choose to use a factor analysis-like covariance matrix of the form  $\mathbf{T} +$  with diagonal. We call this model LDA-FA for Linear Discriminant Analysis with Factor Analysis covariance. The factor analysis covariance model is slightly more general than the PPCA covariance model used by Tipping and Bishop in their LDA experiments. Note that while Tipping and Bishop also consider learning PPCA models with missing data, they do not consider the simultaneous application of PPCA to linear discriminant analysis with missing data.

The factor analysis covariance parameters are learned by first centering the training data by subtracting o the

appropriate class mean as seen in Equation 3, and then applying the Expectation Maximization algorithm for factor analysis with missing data. The dimensionality of the latent factors Q is a free parameter that must be set using cross validation.

#### 6.2.6 Discriminatively Trained LDA and Missing Data

One of the main drawbacks of generatively trained classification methods is that they tend to be very sensitive to violations of the underlying modeling assumptions. In this section we consider a discriminative training procedure for the LDA-FA model described in the previous section. The main insight is that we can t the LDA-FA model parameters by maximizing the conditional probability of the labels given the incomplete features instead of maximizing the joint probability of the labels and the incomplete features. This training procedure is closely related to the minimum classification error factor analysis algorithm introduced by Saul and Rahim for complete data.

The posterior class probabilities given an incomplete feature vector are again given by Equation 1



$$\begin{aligned} \mathcal{L} &= \sum_{n=1}^N \sum_{c=1}^C [y_n = c] \log(P_n^c) \\ P_n^c &= \frac{A_n^c}{\sum_{c'} A_n^{c'}} \\ A_n^c &= \theta_c |2\pi \Sigma^{oo}|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x}_n^o - \mu_c^o)^T (\Sigma^{oo})^{-1} (\mathbf{x}_n^o - \mu_c^o)\right) \end{aligned} \quad \text{----- (3)}$$

### Conditional Maximum Likelihood Estimation

We derive a maximum conditional likelihood learning algorithm for the LDA-FA model in this section. We optimize the average conditional log likelihood with respect to the parameters  $\theta_c$ ,  $\mu_c^o$ , and  $\Sigma^{oo}$ , and using non-linear optimization. We first transform the parameters  $\theta_c$ ,  $\mu_c^o$ , and  $\Sigma^{oo}$  to eliminate constraints.  $\theta_c$  represent the parameters of a discrete distribution with normalization and positivity constraints, while  $\mu_c^o$  simply has to be positive since it is a variance parameter. We use the mappings shown below.

$$\Psi_{ii} = \exp(\phi_{ii}) \qquad \theta_c = \frac{\exp(\omega_c)}{\sum_c \exp(\omega_c)} \quad \text{-----(4)}$$

We begin by computing the partial derivative of the conditional log likelihood with respect to the current posterior class probabilities  $P_n^k$ , and the partial derivative of the posterior class probability with respect to  $A_n^k$ .

$$\frac{\partial \mathcal{L}}{\partial P_n^k} = [y_n = k] \frac{1}{P_n^k} \partial P_n^k \qquad \frac{\partial P_n^k}{\partial A_n^l} = \left( \frac{[k=l]}{\sum_k A_n^k} - \frac{A_n^k}{(\sum_k A_n^k)^2} \right) \partial A_n^l \quad \text{----- (5)}$$

We compute the partial derivative of  $A_n^l$  with respect to  $\omega_c$ , and use the chain rule to find the partial derivative of the conditional log likelihood with respect to  $\omega_c$ . The projection matrix  $\mathbf{H}_n^o$  was introduced in Section 1.2.1. Recall that  $\mathbf{H}_n^o$  projects the observed dimensions of  $\mathbf{x}_n^o$  back into D dimension such that the missing dimensions are filled in with zeros. These projection matrices arise naturally when taking the derivative of a sub-matrix or sub-vector with respect to a full dimensional matrix or vector. Also recall that  $\mathbf{o}_n$  refers to the vector of observed dimensions for data case  $\mathbf{x}_n$  such that  $o_{in} = d$  if  $d$  is the  $i^{\text{th}}$  observed dimension of  $\mathbf{x}_n$ .

	Simple		Mix		Overlap	
	Loss	Err(%)	Loss	Err(%)	Loss	Err(%)
LDA-FA Gen	0.0449	1.75	0.3028	20.50	0.2902	13.50
LDA-FA Dis	0.0494	2.00	0.0992	3.25	0.2886	13.75

Table 6.1: Summary of illustrative results for generative and discriminatively trained LDA-FA models. We report the log loss (average negative log probability of the correct class), as well as the average classification error.

### CONCLUSION

The multiple imputation results show much smoother classification functions than any of the other methods. This results from a combination of noise due to sampling variation in the imputations, as well as from the fact that each classification function results from an ensemble of logistic regression classifiers. The

multiple imputation results also show that multiple imputation can perform well even if the imputation model is incorrect. There is little difference in the classification functions based on a one component factor analysis model, and a two component factor analysis mixture. The reason for this behavior is

explained by the fact that if a single Gaussian is used to explain both clusters in the Simple training data set, the conditional densities are approximately correct for most data cases, even though a two component mixture gives a much better fit to the data.

## REFERENCE

1. Arthur E. Hoerl and Robert W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12(1):55{67, 1970.
2. David W. Hosmer and Stanley Lemeshow. *Applied Logistic Regression*, Second Edition. John Wiley and Sons, 2000.
3. Hemant Ishwaran and Mahmoud Zarepour. Exact and Approximate Sum Representations for the Dirichlet Process. *The Canadian Journal of Statistics*, 30(2):269{283, 2002.
4. Tony Jebara, Risi Kondor, and Andrew Howard. Probability Product Kernels. *Journal of Machine Learning Research*, 5:819{844, 2004
5. M. Jordan. Why the Logistic Function? A Tutorial Discussion on Probabilities and Neural Networks. Technical Report 9503, MIT Computational Cognitive Science, 1995.

