Intrinsically Interpretable Machine Learning In Computer Aided Diagnosis

S. Ghosh, Sreejita

DOI:
10.33612/diss.175627883

Chapter 3

Missing values and Imbalanced classes

> Happiness can be found, even in the darkest of times, if one only remembers to turn on the light.
>
> Albus Dumbledore (J.K. Rowling)

Abstract

This chapter introduces the main technical motivations behind this thesis, namely missing values and imbalanced classes. It describes the types of missingness, the state-of-the-art techniques for handling this issue, and scenarios where the available techniques are simply not good enough. Similarly for imbalanced class problem, we first define the issue, the available techniques for handling it, and where and how these techniques might fail. Next we propose our solutions to circumvent these problems.

3.1 Introduction

Certain aspects of real-life data, especially in the medical sector, hinder the straight-forward application of the existing XAI techniques: these are missing data, heterogeneous measurements, and imbalanced classes. While certain types of missingness are ignorable, some others are not and need to be dealt with systematically. Some classifiers simply fail to work when the dataset contains missing values. Other classifiers however make certain assumptions about the missingness pattern to be able to compute some result, albeit which might be far from representative, due to the assumption about missingness pattern being incorrect. The first two sections of this chapter discusses the different categories of missingness and the possible strategies to handle them. The following two sections describe the issue of imbalanced classes, existing strategies to handle the issue and our proposed strategy.

3.2 Categories of missingness

The aforementioned challenge of missing data can arise due to a variety of reasons. Some causes are completely *random*, such as empty entry by a medical personnel, or

a subject choosing to drop out of a study mid-way. Some causes are more *systematic* such as a sensor being unable to measure values beyond a certain range, or the attending physician not prescribing certain tests for certain patients because the outcome of those tests are thought to be irrelevant. Little and Rubin in (García-Laencina et al. 2010), (Little 1988) have categorised missingness into three types.

The following examples will illustrate the differences. Let χ denote a dataset with D features and N instances and r is a missingness indicator, such that $r_{ij} = 1$ if χ_{ij} is missing and 0 if available. Roderick Little explains in (Little 1988) that (1) function $f(\chi, \Psi)$ where Ψ denotes unknown parameters, and (2) distribution $f(r|\chi, \Psi)$ for r, and indexed by unknown parameters Ψ, together describe a full model for the data and the missing-data mechanism in χ. χ is constituted by χ_{obs} and χ_{miss} where the former denotes observed values of χ and the latter denote the missing values.

3.2.1 Missing completely at random (MCAR)

Rubin, in 1976, defined the missingness to be of type MCAR if $f(r|\chi_{obs}, \chi_{miss}, \Psi) = f(r|\Upsilon)$ for all χ, where r is the missingness indicator variable, Υ is the unknown parameter, and f is probability or density function. That is, missingness would be of type MCAR if the missingness was independent of the observed or missing values of the dataset χ (Little 1988). It indicates if the missingness is neither dependent on the observed nor on the missing values of the dataset $\chi \in \mathbb{R}^{N \times D}$ (Little 1988),(Little and Rubin 2019) A common example of MCAR would be a blood vial of a subject from a study that is accidentally broken resulting in blood parameters being unmeasureable (García-Laencina et al. 2010).

3.2.2 Missing at random (MAR)

Rubin defined missingness to be of type MAR if the missingness is independent of the missing values but likely to be dependent on the observed values, i.e., when $f(r|\chi_{obs}, \chi_{miss}, \Upsilon) = f(r|\chi_{obs}, \Upsilon)$ (Little 1988). An example of such missingness is a sensor occasionally failing to acquire data due to power outage. In this scenario the actual variables where data are missing are the cause of some other external influence, which are recorded (availability of power) (García-Laencina et al. 2010). Here the observed variable is availability of power and the data collected by the sensor are dependent on the availability of power, thus the missingness in the sensor data is dependent on an observed variable, which is availability of power.

3.2.3 Missing not at random (MNAR)

This category of missingness is dependent on the missing values themselves. The cause for this can be systematic, such as the instrument failing to record a parameter when its values are lower than or higher than a certain limit; such data are defined as censored (García-Laencina et al. 2010) . Alternatively, dataset built from differ-

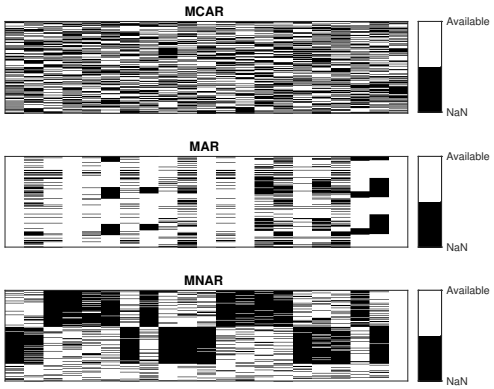

Figure 3.1: The three broad categories of missingness created on the exact same dataset of 20 dimensions and 900 instances.

ent studies can have MNAR type missingness due to different labs not measuring different parameters of subjects.

Figure 3.2.2 illustrates the three categories of missingness. Though MAR might visually appear similar to MNAR, statistical tests will show dependence of missing values on certain dimensions with no complete data. Of the three categories of missingness the first two types are said to be ignorable. However according to (García-Laencina et al. 2010) no established method exists to handle MNAR.

3.3 Missingness handling strategies

(García-Laencina et al. 2010) suggests that there are broadly four different strategies to handle missing data: (1) deletion of incomplete cases and performing classification on the complete samples only, (2) imputation of missing values using observed data, (3) generative modelling of the data distribution, (4) using ML techniques capable of classifying an incomplete dataset.

3.3.1 Case deletion

This technique is applicable when there is enough instances with complete data, from all the classes, or when it is possible to retake the complete measurements of instances with missing values. Even though undoubtedly it is the most straightforward and simplest strategy, this strategy potentially loses a lot of information, especially when many instances with partially observed features exist. Furthermore, we often do not have an abundance of data for analysis in many domains such as Medicine, where any loss of information is undesirable. Additionally, deleting cases with missing values work only when the missingness is ignorable, i.e., type MCAR

or MAR. When the missingness is of type MNAR then removing instances with missing values is likely to introduce bias in the data since the missingness pattern is systematic (Little and Rubin 2019).

3.3.2 Generative modelling of data distribution

(Little 1988) suggests that generative modelling of data is possible when missing data are ignorable. However, this paper suggests that while such models do not require missingness to be MCAR explicitly, they are still more sensitive to model misspecification when the missingness is not MCAR. Therefore unless exploratory analyses of the dataset imply that the missingness type is indeed MCAR this strategy should be used cautiously.

3.3.3 Imputation

In this paragraph we discuss the imputation techniques generally used for handling missingness. In classification tasks of data with missing values it is assumed that only the feature value is unknown but the feature itself is present. Following this assumption the missing values are imputed during pre-processing of data, and then a standard classifier is applied.

Single imputation

Imputing missing attributes with zeros, mean or median of the existing instances of that feature, or with k-nearest neighbours (kNN) applied to existing instances of that features are quite common for missing data of type MAR and MCAR (Chechik et al. 2008). When the missing variable(s) of interest are correlated with the observed variables from complete samples, regression is the appropriate imputation technique. It is better than mean imputation since it preserves the variance and covariance of the features with missing data. This technique however fails when imputing missing values in an independent feature, since regression imputation will cause the imputed missing values of the independent features to be correlated to the rest of the features of the dataset, thus changing the original characteristics of the dataset. Additionally the variance in the dataset is lost when applying this imputation technique (García-Laencina et al. 2010).

Two other categories of single imputation are hot and cold deck imputation. In hot deck imputation the missing components of a data vector are replaced by the corresponding values found in the complete data vector which is closest to the former data vector (whose missing values are being imputed). The disadvantage of this technique is that global properties of the dataset are ignored, since this imputation is based on only the single complete closest data vector. In cold deck method the data source for imputation and the dataset to be imputed cannot be the same (García-Laencina et al. 2010).

Multiple imputation

In multiple imputation (MI) the dataset with missing values is imputed with a set of different *likely* values. MI is also often performed using a regression-based technique called multivariate imputation by chained equations (MICE) (Royston et al. 2011). The suggested technique of regression is essentially a type of hot-deck imputation, performed multiple times. Among the available matching techniques for the hot-deck part, predictive mean matching (PMM) proposed by Little and Rubin, and explained in the following section, is very promising. In MI approaches several imputed datasets are created and the same classifier is applied on each of these datasets. The final decision is made from this ensemble of predictors produced by the classifier trained on the different possible *completed* datasets.

Multiple imputation by chained equations (MICE)

MICE assumes that missingness is of type MAR, and implementing MICE on data with MNAR type of missingness might result in biased estimates. MICE consist of the following steps (Azur et al. 2011).

1. A simple single imputation, such as mean or median imputation is performed on all but one variable (feature). Let us denote it as χ_f. The imputed values are placeholders.

2. The observed values from χ_f in (1) are regressed on the other variables (features).

3. The missing values for χ_f are then imputed with predictions from the regression model.

4. Another variable, previously with placeholders, is selected next, and the placeholders are removed to repeat steps (1)-(3). This time χ_f from step (1) is used as an independent variable in the regression models, with both its observed and the *regression-based* imputed values. This variable from which the placeholder values were just removed now becomes the new χ_f.

5. The above process is performed across all the variables one at a time for a user-defined number of iterations. Thus every variable becomes χ_f once during each iteration. This constitutes one imputed set of data.

6. The above process is repeated for a user-defined number of times to obtain those many imputed sets. In each repetition of the above process a different variable with missing value is selected as the starting χ_f.

We imputed our data using the MICE package in R by (Royston et al. 2011), following the predictive mean matching (PMM) strategy.

MICE with Predictive mean matching

Suppose in the dataset (denoted by χ) the feature vector χ_2 has values missing for some instances (χ_2^{miss}) and has available values in the remaining instances (χ_2^{obs}). Let χ_3 be a feature with no missingness. Estimation of a linear regression of χ_2 is done on χ_3 (for complete cases in χ_2, i.e., χ_2^{obs}) and a set of regression coefficients B are obtained. To ensure variability a number of random draws are made from this posterior predictive distribution of B to form new sets B*. These new sets of B* are used to obtain *predicted* values for all χ_2. Next, a case from χ_2^{miss} is imputed with value from that case in χ_2^{obs}, to which its predicted values was closest (Royston et al. 2011). This process is repeated for different set B. This is also how the need for explicit modelling of missingness mechanism was bypassed. Predictive mean matching (PMM) was selected as the algorithm for imputation for two reasons: (1) to prevent imputation by unrealistic values and values outside the range of available values, and (2) to obviate the need for an explicit model to capture the cause of missingness.

3.3.4 Techniques which could be applied after imputation

After an incomplete dataset has been imputed any classifier, such as k-nearest neighbours, Random Forest, and so on, can be applied on each of the imputed sets.

Random Forest: Leo Breiman in (Breiman 2001) introduced Random Forest (RF), which is an ensemble of decision trees using bootstrap aggregation (Bagging). Decision tree is a rule based model which can be used for both classifiifcation and regression (Kubat 2017). Due to its transparency it is preferred by the medical community. Each decision tree is a weak learner, meaning that by itself a DT is prone to overfitting on validation set. This problem is avoided in Random Forest because of Law of Large Numbers (Breiman 2001). In Breiman's RF the decision trees are unpruned and each tree learns from a different subset of instances. For classification the final decision is given by the majority vote over all the decision trees. In the MATLAB implementation of Tree Bagger which we used for this contribution the randomness is generated by the random subset selection. Even though it is a robust classifier, due to ensembling RF loses some of the transparency of decision trees. It also cannot provide information about the decision boundaries and representative examples of classes. However information about feature importance can still be obtained by computing a median or mean over the feature importance according to all the decision trees. Observations in a predictor variable are permuted randomly and then the error made by the model is computed. If a predictor variable is not important then the permutation of its observation values should not increase the error made by the model. Conversely if the permutation causes the error to be high it implies that the feature is important (Fisher et al. 2019). Random Forest cannot handle missing data

implicitly in its current state. Therefore one needs to apply multiple imputation on a dataset with missing values before Random Forest could be applied on it.

3.3.5 Techniques which can intrinsically deal with missing values

Multiple imputation is expensive with regards to time and memory with increasing amounts of missigness. Especially in a cross-validation setting this is costly, since it needs to be performed for every training set independently to obtain the parameters for imputing the corresponding test set for fair comparison of the generalization error. To avoid imputation of any kind machine learning techniques, which deal with partially observed data were introduced. Prominent examples of strategies are based on generative modelling followed by Linear Discriminant Analysis (LDA), as for example analyzed by (Marlin 2008). These methods show promising results for missing data of ignorable types MCAR and MAR and cannot necessarily be assumed to work well on MNAR. Alternatively, prototype based strategies have recently emerged to deal with data sets containing missing values (van Veen 2016) and (Ghosh et al. 2020). A more recent development is the Probabilistic Random Forest (PRF) (Reis et al. 2018) which can deal with both noisy and missing data. The techniques we developed are similar in capabilities to the PRF, but much more intrinsically interpretable.

Generative modelling strategies

These are often used for (un)supervised data analysis or as preprocessing for partially observed data. When dealing with high dimensional data containing a relatively small number of instances, factor analysis (FA) is often used for structures covariance approximation. FA, which is the most common latent variable model, assumes that a set of *latent* or *unobservable* factors $t_j, j = 1 \dots Q$ are combined to generate χ. FA tries to relate a D-dimensional observed data vector $\boldsymbol{\chi}$ to its corresponding Q-dimensional vector of latent variables $\boldsymbol{t}$ ($Q < D$) (Tipping and Bishop 1999, Marlin 2008). Vectors $\boldsymbol{\chi}$ and $\boldsymbol{t}$ are related by

$$\boldsymbol{\chi} = \Gamma \boldsymbol{t} + \mu + \epsilon \tag{3.1}$$

Conventionally $\boldsymbol{t} \sim \mathcal{N}(0, 1)$ and $\epsilon \sim \mathcal{N}(0, \Psi)$, i.e., both the latent variables and the noise model are Gaussian. The latent variants are also independent of each other by convention. Ψ is a square diagonal matrix. Γ contains the factor loadings and is of dimension $Q \times D$. Therefore the observed variables $\boldsymbol{\chi} \sim \mathcal{N}(\mu, \Sigma)$ where $\Sigma = \Gamma\Gamma^\top + \Psi$. The parameters Γ, Ψ and μ are optimised for a dataset using the expectation maximization (EM) algorithm. This model illustrates the dependencies between the data variables $\boldsymbol{\chi}$ through the latent variables $\boldsymbol{t}$. (Tipping and Bishop 1999),(Marlin 2008),(Severson et al. 2017). In other words, when variables

in the input space are highly correlated, it can be assumed that they have a common source. Additionally FA has a term to explain what was not explainable by the factors, denoted by ϵ_i. Probabilistic Principal Component Analysis (PPCA) is a special case FA, where instead of a diagonal matrix the covariance is simplified by $\sigma^2 I$. Since the covariance matrix is assumed to be spherical, PPCA is rotation-invariant with regards to the observed data (Marlin 2008),(Tipping and Bishop 1999). Note, that classical PCA is a special case of probabilistic PCA where the noise limit or covariance σ is zero. For supervised analysis these generative model strategies are followed by classification, for example with Linear Discriminant Analysis (LDA) (Marlin 2008). Even though LDA can classify data containing missing values, when the dataset is high dimensional or has small sample size, it is preferable according to (Marlin 2008) to use a structured covariance approximation, such as that given by FA and PPCA. Since the medical dataset we had is both high dimensional and had too few samples in certain conditions, we followed the suggestion in (Marlin 2008), and used LDA on the Q-dimensional dataset ($\boldsymbol{t}$), which in addition to being of lower dimension does not contain missingness. We use PPCA instead of classical PCA because the former retains information about the average variance lost per $D - Q$ dimension in the noise model $\sigma^2 I$ (Tipping and Bishop 1999),(Severson et al. 2017). This makes PPCA amenable to missing data, unlike classical PCA. Further interesting information comparing using PPCA and MICE for learning from data containing missing values can be found at (Hegde et al. 2019).

Probabilistic Random Forest (PRF)

This development by (Reis et al. 2018) is an improvement on the Random Forest (RF) classifier. In both RF and PRF the node of each decision tree contains a condition in a specific feature. However contrary to regular RF, instead of an object being propagated to either left or right branch, in PRF the object is propagated to both branches, carrying with it the probability of belonging to each branch ($\pi_i(l)$ and $\pi_i(r)$). In case of multi-class PRF the object will propagate to all the C branches with probability $\pi_i(c), \quad c \in [1, C]$. This uncertainty is computed on the feature value of that object at the node n. PRF converges to RF as the uncertainties become negligible. The algorithm can also deal with able uncertainties but in this thesis only the feature uncertainty aspect has been investigated. We show the working of a two-class probabilistic tree n the following paragraph (Reis et al. 2018).

Suppose all the splitting criteria are determined already. The first split at the topmost node will be based on the $k - th$ feature,i.e, if $\chi_{i,k} < \chi_1$ the object i will propagate to the right branch with probability $\pi_i(r)$ where $\pi_i(r) = F_{i,k}(\chi_1)$, or to the left with probability $\pi_i(l) = 1 - F_{i,k}(\chi_1)$. Therefore the probability of the $i - th$ object to reach a node n located deeper in the tree is the combined probability for

it to have taken all the turns which would lead it there from the source node (Reis et al. 2018). For example, the probability of this object to propagate from the top node twice to the right and once to the left ($n = r, r, l$) is

$$\pi_i(r, r, l) = F_{i,k_1}(\chi_1) \times F_{i,k_2}(\chi_2) \times (1 - F_{i,k_3}(\chi_3)) \tag{3.2}$$

This generalized form of (3.2) to any arbitrary node is

$$\pi_i(n) = \prod_{\rho \in R} F_{i,k_\rho}(\chi_\rho) \times \prod_{\nu \in L} F_{i,k_\nu}(\chi_\nu) \tag{3.3}$$

where R and L are the sets of right and left branches respectively. Since the PRF treats feature values as probability distribution functions (PDFs) it can implicitly handle and represent missing values without making any assumption about the dataset. During both training and test phases when an object with values missing for certain features reach the nodes containing conditions on those features, equal probability of 0.5 is assigned for propagation to both branches. Similarly for a C class problem. For further details we refer you to (Reis et al. 2018).

3.3.6 Prototype-based machine learning methods

Such ML models can intuitively deal with missing data by adapting prototypes and comparing to new data samples based on the observed dimensions only. A powerful family of prototype based classifiers is based on the concept of Learning Vector Quantization (LVQ), which follows a Nearest Prototype Classification (NPC) scheme, where a new vector is assigned the class label of the prototype to which it is closest. Assume the data consist of N samples $\boldsymbol{x}_i \in \mathbb{R}^D$ accompanied by labels y_i denoting one of C classes and let $\boldsymbol{w}_j \in \mathbb{R}^D$ denote one of C prototypes with labels $c(\boldsymbol{w}_j)$. Now, Generalized LVQ (GLVQ) performs a supervised training procedure aimed at minimizing the following cost function (Sato and Yamada 1996), which exhibits a large margin principle (Hammer et al. 2005):

$$E = \sum_{i=1}^{S} f\left(\mu_i\right), \text{where } \mu_i = \frac{d_i^J - d_i^K}{d_i^J + d_i^K} \ . \tag{3.4}$$

Here the dissimilarity of each data sample $\boldsymbol{x}_i$ to its nearest correct prototype with $y_i = c(\boldsymbol{w}_J)$ is defined by d_i^J and by d_i^K for the nearest wrong prototype ($y_i \neq c(\boldsymbol{w}_K)$). f is a monotonic function and we set to the identity ($f(a) = a$) throughout this contribution. Extensions to GLVQ introduced parameterized dissimilarity measures, such as the quadratic form:

$$d_i^L = (\boldsymbol{x}_i - \boldsymbol{w}_L)^\top \Lambda (\boldsymbol{x}_i - \boldsymbol{w}_L) \quad \text{with } \sum_i \Lambda_{ii} = 1 \ , \tag{3.5}$$

with the positive semi-definite matrix $\Lambda \in \mathbb{R}^{D \times D}$ containing additional parameters for optimization. This led to a family of relevance and matrix extensions (GRLVQ and GMLVQ) that provide intrinsic interpretability in form of relevance of the features for classification determined by the diagonal of Λ (Hammer and Villmann 2002),(Schneider et al. 2007),(Schneider et al. 2009) and discriminant visualization using low-rank decomposition of Λ (Bunte et al. 2012). In (Ghosh et al. 2017) the authors introduced two variants of Generalized Matrix LVQ (GMLVQ) that can deal with missing values. The first variant called NaN-GMLVQ bases on the intuitive idea that one can update the prototypes $\boldsymbol{w}_L$ and matrix Λ in the observed dimensions only for each training sample $\boldsymbol{x}_i$. Accordingly, a new sample is classified with the label of the closest prototype computing the distance Eq. (3.5) without the missing dimensions. The distances are differently scaled dependent on how many features are missing within a sample. However, this does not pose a problem since for training and classification the samples are never compared to each other, but only to the model parameters, which do not contain any missing values. This approach is quite straightforward and works quite well when missing data is of type MCAR. However, when the number of missing features varies a lot across classes or is of type MNAR the prototypes can be adversely affected, being pushed away by a close class that contain samples with more observed dimensions. Countering these served as motivation for the development of an LVQ method that classifies on the hypershpere, instead of Euclidean space, based on an angular dissimilarity measure (ALVQ) as detailed in section 5.2.1.

Developed technique: Angle LVQ

The set of angle-based intrinsically interpretable models we developed were able to handle even systematic missingness, as explained in chapter 5.

3.4 Imbalanced classes

In many domains we face the situation that occurrences of instances from different classes vary in frequency and, on top of it, experts are often most interested in samples of the minority class(es). In the medical field for example, while it is promising that there are more healthy subjects than reported patients, this fact generally poses a challenge in training machine learning models. The issue of class imbalance is even more pronounced when the investigated conditions are rare diseases. The difficulty with training a classifier on a dataset with class imbalance is that many classifiers tend to become biased towards the majority class, due to the probable absence of the minority class samples during training. Additionally, performance measures might also be affected, such as looking at one accuracy value summed over all classes becomes misleading. Literature suggests that most prominent strategies to

handle imbalanced data comprise of bagging, boosting, and sampling, including undersampling and oversampling (Parsons 2005).

3.4.1 Sampling

Sampling involves either stratified sampling, density-based sampling, undersampling of the majority class(es) or oversampling of the minority class(es). (Chawla et al. 2002) explains an oversampling strategy involving synthesis of new artificial training samples belonging to the minority classes to increase the sample size of those classes. In (Ghosh et al. 2017) we observed that undersampling had an adverse effect on the performances of the classifiers we compared. This might have been due to loss of information associated with undersampling.

3.4.2 Bagging

Bagging or Bootstrap aggregation comprises of voting of class-labels by weak or base-learners each of which were trained on slightly different subsets of training data (drawn with replacement). Using bootstrap k different subsets of data were generated from the original training set χ. A weak or unstable learner such as a decision tree is trained on these k subsets of χ. During testing or validation the decision by each of these k learners are (a) put to vote, in case of classifiers, or (b) averaged, in case of a regressor (though for a more robust regression Briemann suggest median instead of mean) (Breiman 1996) (Parsons 2005). For dataset with imbalanced classes instead of the regular bootstrapping, (Fernández et al. 2018) suggests applying data resampling on the bootstrapped samples prior to application of the weak learners. The resampling can be a stratified sampling by taking the class labels of samples into account or oversampling of minority class (OverBagging) (Fernández et al. 2018).

3.4.3 Boosting

While in bagging k subsets of data are generated randomly, in boosting the complementary weak learners are trained on such a subset of dataset which is more likely to contain mostly the samples which were misclassified (Parsons 2005). That is, boosting favours of 'learning from mistakes'. When learning with imbalanced classes boosting can be modified in such a way that minority class samples are assigned higher probabilities of getting selected for belonging to more data subsets on which weak learners are trained and misclassification of minority class samples are heavily penalized. For further details on boosting please refer to (Breiman 2001) and chapter 17 of (Parsons 2005).

3.4.4 Geodesic SMOTE

A well known oversampling method is Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al. 2002). It increases the sample size of the minority classes by randomly creating artificial new training samples between k nearest neighbours of the same class. In (Ghosh et al. 2017) the authors introduced a geodesic variant of the original SMOTE, which synthesized samples on the hypersphere instead of Euclidean space. Therefore an important tool of Riemannian geometry is used, which is the exponential map (Fletcher et al. 2004, Wilson et al. 2014). The exponential map has an origin G, which defines the point for the construction of the tangent space τ_G of the manifold. Let ζ be a point on the manifold and $\hat{\zeta}$ a point on the tangent space. Then, $\hat{\zeta} = \text{Log}_G\zeta$, $\zeta = \text{Exp}_G\hat{\zeta}$ and $d_g(\zeta, G) = d_e(\hat{\zeta}, G)$ with d_g being the geodesic distance between the points on the manifold and d_e being the Euclidean distance on the tangent space. Log and Exp denote a mapping of points from the manifold to the tangent space and vice versa. As described in (Ghosh et al. 2017) we present a point $\boldsymbol{x}$ from class c on the unit sphere with fixed length $|\boldsymbol{x}| = 1$. The point $\boldsymbol{x}$ becomes the origin of the map and the tangent space. Next k nearest neighbours of the selected sample $\boldsymbol{x}$ are found from the same class as the selected sample ($\boldsymbol{x}_\psi \in \mathcal{N}_{\boldsymbol{x}}$) using the geodesic distance between the vectors $\theta = \cos^{-1}(\boldsymbol{x}^\top \boldsymbol{x}_\psi)$. Each random neighbour $\boldsymbol{x}_\psi$ is then projected onto that tangent

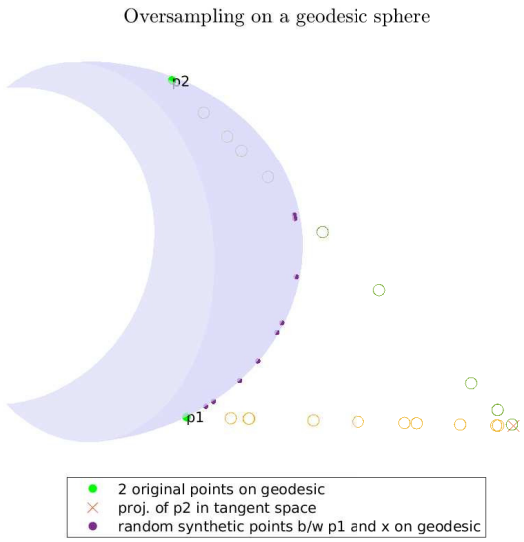

Figure 3.2: Oversampling using SMOTE on a geodesic sphere.

space using only the available features and the Log_G transformation for spherical manifolds:

$$\hat{\boldsymbol{x}}_\psi = \frac{\theta}{\sin\theta}(\boldsymbol{x}_\psi - \boldsymbol{x}\cos\theta). \tag{3.6}$$

Finally a synthetic sample is produced on the tangent space using the same formula as the original Euclidean SMOTE $\hat{\boldsymbol{s}} = \boldsymbol{x} + \alpha \cdot (\hat{\boldsymbol{x}}_\psi - \boldsymbol{x})$, where $\alpha \in (0, 1]$. The new angle $\hat{\theta} = |\hat{\boldsymbol{s}}|$ is then used to project the new sample back to the unit hypersphere by the Exp_G transformation:

$$s = \boldsymbol{x}\cos\hat{\theta} + \frac{\sin\hat{\theta}}{\hat{\theta}}\hat{s} \tag{3.7}$$

This procedure of synthetic sample generation is repeated with other random samples from the class until the desired number of training samples is reached. We proposed to oversample each of the minority classes in the training set until they are equivalent in size to the majority class. This avoids the original SMOTE hyperparameter, namely the percentage of oversampling for each minority class.

3.4.5 Incorporation of expert-knowledge: modified boosting

In order to incorporate knowledge of domain experts into the classifier, a modification was made to the newly developed angular variant of LVQ, such that certain misclassifications were more heavily penalized than certain others. This concept is very similar to that of boosting. This classifier is explained in (Ghosh et al. 2017).

3.5 Conclusion

The existing techniques for handling missing data are capable of handling ignorable (MCAR and MAR) type of missingness. However they are not suitable for MNAR. Probabilistic PCA followed by LDA can deal with missing values intrinsically, however it is not well suited for MNAR type of missingness. Therefore we need classifiers such as probabilistic PRF which can intrinsically handle missing data of any type. Additionally we are interested in classifiers which are explainable. This motivated us to develop and introduce a few variants of LVQ. Even though we also developed a variant of LVQ which could use a boosting-like mechanism to handle imbalanced classes we preferred to use the model agnostic technique of oversampling with the geodesic version of SMOTE based on our experiments in (Ghosh et al. 2017).