Beyond the Boundaries of SMOTE

A Framework for Manifold-based Synthetic Oversampling

by

Colin Bellinger

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the Ph.D. degree in
Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

© Colin Bellinger, Ottawa, Canada, 2016

Abstract

Within machine learning, the problem of class imbalance refers to the scenario in which one or more classes is significantly outnumbered by the others. In the most extreme case, the minority class is not only significantly outnumbered by the majority class, but it also considered to be rare, or absolutely imbalanced. Class imbalance appears in a wide variety of important domains, ranging from oil spill and fraud detection, to text classification and medical diagnosis. Given this, it has been deemed as one of the ten most important research areas in data mining, and for more than a decade now the machine learning community has been coming together in an attempt to unequivocally solve the problem.

The fundamental challenge in the induction of a classifier from imbalanced training data is in managing the prediction bias. The current state-of-the-art methods deal with this by readjusting misclassification costs or by applying resampling methods. In cases of absolute imbalance, these methods are insufficient; rather, it has been observed that we need more training examples. The nature of class imbalance, however, dictates that additional examples cannot be acquired, and thus, synthetic oversampling becomes the natural choice.

We recognize the importance of selecting algorithms with assumptions and biases that are appropriate for the properties of the target data, and argue that this is of absolute importance when it comes to developing synthetic oversampling methods because a large generative leap must be made from a relatively small training set. In particular, our research into gamma-ray spectral classification has demonstrated the benefits of incorporating prior knowledge of conformance to the manifold assumption into the synthetic oversampling algorithms.

We empirically demonstrate the negative impact of the manifold property on the state-of-the-art methods, and propose a framework for manifold-based synthetic oversampling. We algorithmically present the generic form of the framework and demonstrate formalizations of it with PCA and the denoising autoencoder. Through use of the helix and swiss roll datasets, which are standards in the manifold learning community, we visualize and qualitatively analyze the benefits of our proposed framework. Moreover, we unequivocally show the framework to be superior on three real-world gamma-ray spectral datasets and on sixteen benchmark UCI datasets in general. Specifically, our results demonstrate that the framework for manifold-based synthetic oversampling produces higher area under the ROC results than the current state-of-the-art and degrades less on data that conforms to the manifold assumption.

Acknowledgements

As I put the final touches on this dissertation, I cannot help but also reflect upon all that it has meant to arrive at this point and be grateful to those whose wisdom, encouragement, inspiration and quiet support has undeniably facilitated my successes. Frist and foremost, I would like to express my profound gratitude to my thesis supervisor, Dr. Nathalie Japkowicz, who initially sparked my interest in machine learning and earned my respect through her teaching during my Masters, and who has continued to inspire and challenge me throughout my Ph.D. I would also like to express my great appreciation to Dr. Christopher Drummond for joining as my co-supervisor, engaging me in philosophical discussions, whilst moving the process forward and helping to finetune my thesis. I thank my examining committee for contributing their time and offering their wisdom and insights. I must thank all the professors, students and administrators in the School of Electrical Engineering and Computer Science, whom I have had the pleasure of calling my peers. I would like to thank Dr. John Oommen who, amongst many other things, honed what can only be called my very rough skills during my Masters and inspired in me the confidence to enter into my Ph.D.

I have been most fortunate throughout my Ph.D. to have had the opportunity to work with and learn from a great number of researchers outside of the University of Ottawa. I would like to thank Dr. Stan Matwin and Dr. Stenio Fernandes for facilitating my research exchange to Brazil, and my dear friend Dr. Cesar Astudillo for inviting me to Chile to collaborate with him and engage his colleagues in many intriguing discussions. More importantly, I would like to thank Cesar for inspiring me with his passion for details, and his omnipresent thirst for knowledge and understanding. Much of my research would undoubtedly have been impossible without the thoughtful contributions of Dr. Kurt Ungar and Rodney Burg at Health Canada. I am endlessly thankful for all that they have contributed.

I am ever grateful to my office-mates Vincent Barnab-Lortie and Adrian Taylor for their collaborations and sharing their inspiring research with me during their time in our little office, and to my good friend Shiven Sharma for the many hours of discussion and editing that he generously offered, not to mention his dedication to introducing me to authentic Indian food. To my dear friends in the triathlon, cycling and running communities, I express my great appreciation for your encouragement and partaking in many hours of mental rejuvenation while swimming, biking and running around Ottawa and in Gatineau Park. I would like to thank my lovely girlfriend, Lauren Stoymenoff,

who endured many hours of thesis talk and was always there to offer moral support and encouragement. Her contribution was one of absolute importance.

Finally, I would like to thank the University of Ottawa, Dr. Japkowicz, the Province of Ontario and the Radiation Protection Bureau at Health Canada for the financial support that has facilitated my doctoral studies.

Dedication

To my endlessly supportive parents Deborah and Larry Bellinger

Live the full life of the mind,
exhilarated by new ideas,
intoxicated by the Romance of unusual
– Ernest Hemingway

Contents

List of Tables

List of Figures

Chapter 1

Introduction

One might be excused for wondering if the classical statistical inference theorem of *no-free-lunch* [Wolpert, 1996], though still referenced in passing, has lost its relevance in the minds of many in the machine learning community. Wolpert wisely informs us that no algorithm uniformly outperforms all others for all datasets, yet so many researchers design in the grandios spirit of no-assumptions-necessary. Indeed, the motivation of many is to design a learning algorithm that is superior across all domains. We liken this objective to the pursuit of a universal cancer drug. Whilst we can all see the merits in it, practically speaking we are much more successful when we develop treatments for individual forms of cancer, such as Hodgkins lymphoma or Acute lymphoblastic leukemia.

The no-free-lunch theorem results from the fact that each classifier has an implicit bias that is necessary to facilitate learning. It is the relationship between the bias and the latent properties of the target domain that renders the learner more or less suitable for the task. Therefore, since we have a large number of latent properties in the universe of datasets, we also need a large set of algorithms with diverse biases. It is when we appropriately pair the bias of the learner with the properties of the target domain that the classification performance reaches its full potential.

Indeed, in our practical experience, we have found scenarios where making a simple and appropriate assumption regarding the properties of the data leads to the selection of a method that outperforms those algorithms, such as support vector machine (SVM) and multilayer perceptron (MLP), that are our best attempts at contradicting the no-free-lunch theorem [Sharma et al., 2012a, Sharma et al., 2012b]. This highlights our belief that it is important to design and apply learning algorithms with biases that are appropriate for the target data whenever possible. This relationship between learning

algorithms and the target data was identified even earlier in [Mitchell, 1980]. In this thesis, we demonstrate that it is equally, if not more, important to apply this notion when selecting synthetic oversampling algorithms for data suffering from class imbalance. To this end, we produce a comprehensive set of experiments that show that designing and applying synthetic oversampling methods with underlying biases that are appropriate for the target data, leads to the induction of better classifiers.

The metamorphosis of machine learning as a discipline isolated in the neon lit labs of computer scientists to data science where the path forward is illuminated by both our advanced algorithms and the knowledge, methods and expertise of the target domain has facilitated solutions that are both ingenious and practically founded with an emphasis on the properties of the data at hand. We argue that it is now, perhaps more than ever, true that the machine learning community can, and should, design and apply learning algorithms with biases that are appropriate for the general properties of the data. This does, indeed, occur in some enclaves of machine learning. It is, for example, commonly expected that deep learning methods are most appropriate for data with an implicit, embedded hierarchical structure whereas SVMs are believed to be superior on flat data. Likewise, if we know the distributional property of the data, then we should utilize it. With this in mind, we argue that the same informed selection should be utilized when it come to synthetic oversampling methods.

In this thesis, we narrow our general claim regarding the necessity of using domain properties to inform the selection of machine learning algorithms by considering it within the context of synthetic oversampling for imbalanced classification. Specifically, we examine which algorithmic form is most appropriate for synthetically oversampling data that conforms to the manifold property and propose a framework based on our findings.

Our focus is motivated by the importance of manifold learning and class imbalance in machine learning. Class imbalance is widely held to be one of the essential problems to address in machine learning [Yang et al., 2006]. To this end, the machine learning community has come together to address the matter in a number of high profile workshops, conferences and special issues. These include the American Association for Artificial Intelligence workshop on Learning from Imbalanced Data Sets [Japkowicz, 2000], the International Conference on Machine Learning workshop on Learning from Imbalanced Data Sets [Chawla et al., 2003] and the Association for Computing Machinery Special Interest Group on Knowledge Discovery and Data Mining Explorations [Chawla et al., 2004]. We are particularly interested in cases where minority class training data is rare and unrepresentative; this is referred to as absolute imbalance. Identifying and managing

it is essential for good classification accuracy [He and Garcia, 2009]. Similarly, manifold learning is seen to be integral to achieving good performance on high-dimensional machine learning problems [Mika et al., 1999, Roweis and Saul, 2000, Zhang and Zha, 2004, Hinton and Salakhutdinov, 2006].

Data that conforms to the manifold property is typically high-dimensional, with the probability density residing in a lower dimensional space. A common and effective analogy for understanding manifolds is to think of sticky rice on a plate. Looking down on the rice, we can specify the location of each grain using a two-dimensional coordinate system on the plate. If we raise the plate off the table and tilt it, we would typically specify the location of each grain in the three-dimensional space represented by the room. Nonetheless, nothing has changed for the rice with respect to the plate. Therefore, we could still use the plate-based, two-dimensional coordinate system for the grains if we knew the orientation of the plate in the three-dimensional space. From a machine learning perspective, the embedded-space is typically the ideal representation of the data because it is lower-dimensional and focused on the density of the data. Manifold learning algorithms infer the lower dimensional space from the higher dimensional feature space of the training data. In doing so, they increase the likelihood of inferring a good model for the purpose of classification or other machine learning applications. It is, therefore, of significant benefit to identify or infer conformance to the manifold property and apply a machine learning method that is appropriate for such data.

Within manifold learning communities certain domains, such as text and image, are seen as primary examples of conformance to the manifold assumption. These are not necessarily hard facts that are mathematically tested nor explicitly extracted from domain experts, but rather, they are the result of thoughtful consideration of the data itself. By thinking about the features in face recognition (the pixels in the image) and the fact that the target class is constructed from a combination of a subset of the pixels, for example, we can infer that the domain of face recognition conforms to the manifold property.

Through discussions with our colleagues at the Radiation Protection Bureau at Health Canada regarding the nature of their gamma-ray spectral classification task and the physical properties of gamma-ray spectra, we were able to apply similar logic to infer soft conformance to the manifold assumption in their domain. In particular, we know that the radioactive occurrences that form the minority class will affect subset specific energy levels in the spectra, thereby producing a subspace that is formed from a, perhaps complex, combination of the original features.

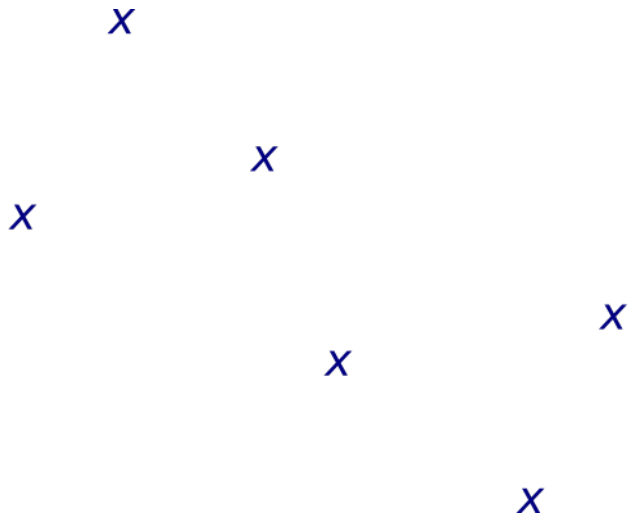


Figure 1.1: Training instance.

Synthetic oversampling algorithms are particularly affected by the no-free-lunch theorem because they are required to take a large generative leap from a small number of examples; by extension, the overall performance of synthetic oversampling can be positively impacted by incorporating a bias that is appropriate for the properties of data.

As a mental exercise, consider the induction of a function to represent the training points in Figure 1.1. If we have no prior knowledge regarding the properties of the target distribution, then we can do little more than apply an arbitrary generative bias. In such a case, assuming a linear function with additive noise is likely our best bet. Given inside information regarding a property of the generating function, however, we can do much better. Say, for example, that the latent distribution of the points in Figure 1.1 involves a sine function. We can use this information accordingly and estimate a more accurate representation of the generating function; this is demonstrated in Figure 1.2.

Synthetic minority oversampling methods artificially balance the prior probability of the minority portion of the training set with the majority portion of the training set in order to reduce the biasing effect of imbalance on the induced classifier. Existing methods have been shown to be effective for dealing with problems of class imbalance [Chawla et al., 2002, Gao et al., 2012, Fecker et al., 2013, Shao et al., 2014] in some circumstances. However, developers and practitioners have failed to take *a-priori* knowlege of the properties of the target data into account. As we demonstrated in the example above, to ignore prior knowledge when it is available is to put an unnecessarily low ceiling on performance.

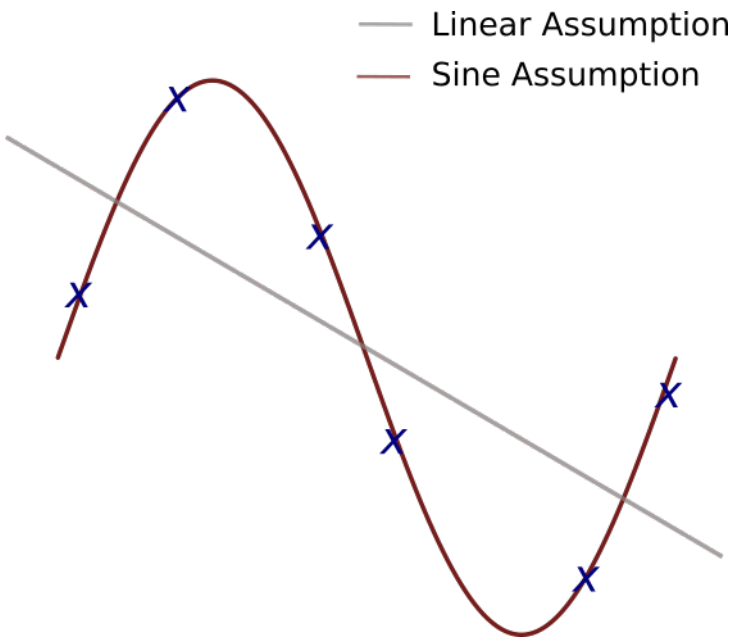


Figure 1.2: Approximating a sine function with and without prior knowledge.

This thesis utilizes the fact that for a large number of machine learning domains, we have *a-priori* knowledge, or a good assumptions, regarding conformance to the manifold property. Based on this knowledge, we devise and demonstrate synthetic oversampling methods that outperform the state-of-the-art methods on imbalanced classification tasks when the data conforms to the manifold property.

1.1 Motivation

The story of this thesis commences with our efforts to perform gamma-ray spectral (GRS) classification for the Radiation Protection Bureau at Health Canada, where the classification task is imbalanced. Class imbalance has been shown to have a negative impact on the performance of induced binary classifiers; this is particularly the case when the underlying distribution is complex and when the minority class is rare [Weiss, 2004, He and Garcia, 2009]. Indeed, it has been recognized as one of the ten challenging problems in data mining research [Yang et al., 2006].

The gamma-ray spectral data that motivated this work conforms to the manifold assumption [Chapelle et al., 2006]. Specifically, the probability density can be said to live in a lower-dimensional space. Thus, the target data is:

- Imbalanced:
- High-dimensional; and
- Conforms to the manifold assumption.

Data that is imbalanced and high-dimensional undoubtedly forms a very challenging classification task.

We believe that the manifold property holds the key to solving many high-dimensional imbalanced classification tasks because it enables us to operate in the implicit manifold space that better represents the domain. Moreover, by mapping the data to a lower dimensional manifold space, we are reducing the impact of the curse-of-dimensionality, which indicates that in high-dimensional spaces data is inherently sparse [Bellman, 1961, Hughes, 1968]. Lowering the dimensionality, therefore, reduces the sparseness and increases the likelihood of producing a good representation of the data from which representative samples can be generated.

In many high-dimensional machine learning problems, such as classification and clustering, the key to success is believed to reside in discovering said space. As a result, a significant amount of research has been devoted to the derivation of methods capable of inferring the manifold-space from a training set [Mika et al., 1999, Roweis and Saul, 2000, Zhang and Zha, 2004, Hinton and Salakhutdinov, 2006]. This fact, however, has gone largely unconsidered by methods designed to cope with class imbalance. As a result, the state-of-the-art methods of synthetic oversampling underperform on data that conforms to the manifold property by under-generalizing along the manifold and overgeneralizing away from the manifold. Thus, we are motivated to develop an appropriate method of synthetically oversampling for gamma-ray spectral data, and all other domains that conform to the manifold assumption. We argue that this will mitigate the negative impacts of class imbalance in the target domain.

1.2 Contribution

At the highest level, our contribution is to emphasize the importance of designing and applying algorithms with assumptions and biases that are appropriate for the properties of the target data. We are particularly interested in cases of class imbalance in which the data conforms to the manifold assumption. To this end, we aim to show that by ignoring the presence of the manifold property, existing synthetic oversampling methods fail to

achieve their full potential. In accordance with this newly accentuated fact, we devise and deploy a framework for synthetically oversampling manifold data, and demonstrate its superiority on artificial datasets, benchmark datasets and three real-world gamma-ray spectral datasets.

Our primary contribution is made to the field of class imbalance. In particular, we have formalized the previously unconsidered fact that, given the large generative leap required of synthetic oversampling methods, it is absolutely beneficial to incorporate knowledge about the properties of data domain whenever possible. We are particularly interested in the presence of the manifold property, and as such, we have studied its impact on synthetic oversampling.

We show that the state-of-the-art methods of synthetic oversampling are negatively impacted by the presence of the manifold property in terms of weakened area under the ROC curve (AUC) results [Provost and Fawcett, 1997]. Thus, we argue that we are currently not achieving our full potential with reliance on these limited methods. To this end, we propose a framework for manifold-based synthetic oversampling and demonstrate its superiority on artificial datasets, benchmark datasets and three real-world gamma-ray spectral datasets. Therefor, we list our formal contributions as:

- Unequivocally show the value of understanding which learning biases are best suited for specific data properties, such as deep nets on hierarchical data and manifold learning on manifold data.

- Empirically demonstrate the substantial impact of applying generic synthetic oversampling methods when the manifold assumption holds.

- Develop an appropriate framework for synthetically oversampling data that conforms to the manifold assumption.

In order to evaluate our framework, we have devised a method of augmenting benchmark datasets to conform to the manifold assumption with varying levels of significance in order to independently test the impact of the manifold property on synthetic oversampling. In addition, we demonstrate how to use the scree test [Cattell, 1966a] to test the prominence of the manifold in the target data. We find that the manifold property has minimal impact on classification when the latent distribution has a low level of complexity in terms of class separability [Li, 2006]; however, for domains with a high degree of complexity, even weak conformance to the manifold property can have a major, negative impact on synthetic oversampling.

1.3 Organization of Thesis

The remainder of this thesis is structured as follows: the subsequent chapter, Chapter 2, considers the related work. Specifically, the topics of class imbalance and autoencoders are discussed. Chapter 3 presents the state-of-the-art in synthetic oversampling and demonstrates the negative impact of the manifold property on these algorithms. This chapter accentuates the implicit limitation of the methods that are currently prominent in the literature and sets the scene for the development of a more appropriate framework of algorithms for synthetic oversampling. In Chapter 4, we propose a general framework for synthetically oversampling data that conforms to the manifold, and illustrate its superior performance with two distinct manifold learning techniques. Finally, our concluding thoughts are presented in Chapter 5.

Chapter 2

Related Work

2.1 Introduction

The essential assumption in machine learning is the availability of a representative set of training instances drawn from each category in the target domain. In problems of class imbalance, however, this assumption is violated and traditional classifiers can suffer from a sort of *black swan* mentality where their overly biased preference is based on their limited experience rather than informed prediction. As a result, the class imbalance problem has long been recognized as one of the most important problems to address in data mining [Yang et al., 2006].

Class imbalance has been shown to take on a variety of forms, and to vary significantly in its complexity [Weiss, 2004, He and Garcia, 2009]. Fundamentally, however, in binary classification, class imbalance refers to the scenario in which one class has significantly fewer training examples than the other class. Class imbalance need not be limited to a binary setting. For simplicity, however, we will proceed to refer to it in the binary context, whilst understanding that the issue extends to multi-class domains.

Class imbalance can lead to the induction of a classifier that is heavily biased in favour of the majority class. In the most extreme case, we can image a naïve classifier learning to always predict to majority class.

In class imbalance, the ratio between classes commonly ranges between $1:100$ to $1:10,000$ or more. The imbalance itself may result from an intrinsic imbalance which naturally belongs to the domain or an extrinsic imbalance resulting from sampling, cost, time, storage limitations, *etc* [He and Garcia, 2009]. He and Garcia describe two scenarios in class imbalance that have fundamentally different impacts on classification. Cases of

relative imbalance involve class $\omega_{minority}$ being dominated in quantity by class $\omega_{majority}$. This, they argue, does not necessarily impact learning because the training instances in the minority class can be representative in spite of the fact that they are outnumbered. Alternatively, absolute imbalance imbalance refers to rarity. The imbalance is such that there is a lack of representative data in the minority class. This makes learning difficult regardless of the $\omega_{minority} : \omega_{majority}$ ratio.

Prominent data domains, such as scientific, environmental and machine sensor classification often involve the challenging characteristic of class imbalance combined with high-dimensionality. In many cases domains of this nature are of a particular form of the general problem that is characterized by a high-dimensional data-space in which the data probability density resides in a lower-dimensional manifold.

As previously stated, the objective of this thesis is to introduce a framework for synthesizing samples for imbalanced classification problems that conform to the *manifold assumption*. This chapter serves to:

- Motivate the use of synthetic oversampling for managing class imbalance based on the current literature;
- Demonstrate that the current state-of-the-art in synthetic oversampling methods do not account for the manifold property; and
- Justify the merit of the proposed framework for synthetically oversampling the manifold by highlighting the successful application of manifold-based methods in other areas of machine learning.

2.2 Class Imbalance

Class imbalance appears in a wide variety of important and challenging domains ranging from remote sensing, oil spill, machine fault, and fraud detection [Williams et al., 2009, Kubat et al., 1998, Fecker et al., 2013, Provost and Fawcett, 2001] to credit and text classification, learning word pronunciations and many medical domains [Zhang and Wang, 2011, Lewis et al., 1994, van den Bosch et al., 1997, Cohen et al., 2006]. A major impact of learning in this setting is that classifiers often suffer from poor performance in terms of low recall [Akbani et al., 2004].

The primary methods of coping with class imbalance are via sampling the training set and cost adjustment; sampling methods adjust the class distribution in the training set

and cost-based methods serve to bias the classifier by skewing the cost associated with misclassifying instances of the minority class. In this chapter, the state-of-the-art in class imbalance is presented and their weakness are emphasized in relation to the proposed method.

To facilitate the discussion of the need for an alternate synthetic oversampling method, we divid the methods of coping with class imbalance into those that perform some form of oversampling and those that do not. We commence by focusing our attention on those methods that do not perform some form of oversampling in order to demonstrate that, in spite of the fact that they can indeed be helpful in some cases, oversampling is necessary. Specifically, when the minority training set is small relative to the complexity of the data distribution, these methods risk:

1. Removing informative and important instances from the majority class; and,

2. Overfitting the minority class.

After we have addressed undersampling and the cost-based methods, we proceed to oversampling, with a particular focus on methods that synthesize new instances for training. We demonstrate that although the discussed methods are explicitly intended to avoid the perils of information loss and overfitting, they do not account for the manifold property. The result of this is that when the data conforms to the manifold assumption, they over-generalize orthogonally to the manifold and under-generalize along it.

In the final section of the chapter, we illustrate that though the manifold property has been neglected in this sub-domain of class imbalance, it has been successfully addressed in most other areas of machine learning.

2.2.1 Undersampling

At a highly granular level, it can be said that there exists a set of circumstances in which methods that do not perform some form of oversampling are insufficient for dealing with class imbalance. This is, indeed, the case when the degree of imbalance is high and when the majority class is complex.

Random undersampling is one of the simplest ways to deal with class imbalance. It adjusts the class distribution by producing a training set $S = \{S_{min} \cup E\}$, where E is a set of random samples drawn without replacement from S_{maj}. The weaknesses of this method are well documented. In particular, random undersampling is know to suffer from high variance [Wallace et al., 2011] and has the potential to discard valuable information

from the majority class, whilst still risking overfitting the minority class [Han et al., 2005]. In particular, random undersampling can cause the removal of a large number instances from the training set. In doing this, it can create small disjuncts, which are areas that are correctly classified by just a few instances. Classification algorithms have been shown to be error prone on small disjucts [Holte et al., 1989]; thus, in addressing class imbalance with random undersampling, we can actually produce a degradation in performance. This is particularly the case when the minority training set is very small, thus necessitating extensive undersampling of a possibly complex distribution [Akbani et al., 2004].

As a result of the weaknesses of random sampling methods, a large number of heuristic-based sampling methods that rely on notions of helpful and unhelpful training instances have been proposed to guide sampling in a more systematic way. When it comes to undersampling, these methods attempt to remove noisy and borderline instances and/or majority class instances that reside far from the border. The former category of methods presume that removing borderline instances makes it easier to induce a good boundary, whilst the latter remove instances at a great distance from the border because they are assumed to add little value during induction. It can be seen that these methods take very different positions regarding which training instances are most helpful. Removing borderline instances will help shift the decision boundary towards the centre of the majority class for some classifiers, thereby reducing bias and resulting in a more general solution. When the shift occurs, the risk is that important subconcepts will be lost. Alternatively, removing instances far from the border may not help shift the decision boundary. In this case, the risk of undersampling important subconcepts is reduced, but the classifier may remain overly biased towards the minority class. The general usefulness of these methods is dictated by the properties or the target data and the implicit bias of the selected classifier; there is, indeed, a dearth of research into the specifics of this relationship [Wallace et al., 2011]. We certainly encourage more research into this area. From the perspective of data that conforms to the manifold property, these fail because they do not take a geodesic approach that would enable them to measure distance along the surface of the manifold, which is made explicit in the next paragraphs.

Tomek links [Tomek, 1976] is, perhaps, the best known method for cleaning noisy and borderline instances. A pair (x_i, x_j) is considered to be a Tomk links if $\neg\exists x_k$ s.t. $dist(x_i, x_k) < dist(x_i, x_j)$ nor $dist(x_j, x_k) < dist(x_i, x_j)$ and $x_i \in S_{min}, x_j \in S_{maj}$. If x_i and x_j form a Tomek links, then either x_i or x_j is noise or they are on the class border.

Wilson's editing nearest neighbour [Wilson, 1972] can also be applied to under-

sampling. This is done by removing the subset of instances in S_{maj} for which at least two of their three nearest neighbours belong to the minority class [Laurikkala, 2001]. In general, majority class instances that are close to the minority class instances are removed. For more aggressive undersampling, it can be modified to remove the $x_i \in S_{maj}$ if the labels of $kNN(x_i)$ for $k = 3$ disagree, and remove the $kNN(x_i)$ for $x_i \in S_{min}$ if $kNN(x_i) \in S_{maj}$.

Three nearest neighbour-based methods referred to as *near miss* can be used to select majority class methods to undersample from the border region [He and Garcia, 2009]. In the first, each instance $x_i \in S_{maj}$ whose mean distance to its kNN is the smallest is removed. Whereas for the second the $x_i \in S_{maj}$ whose farthest neighbours are the closest are removed. In the third, the $kNN(x_i)$, where $x_i \in S_{min}$ and the $kNN(x_i) \in S_{maj}$, are removed.

Both the condensed nearest neighbour rule [Hart, 1968] and the so-called most-distant method [He and Garcia, 2009] remove instances that are far from the border, and thus deemed to be uninformative, from the majority class. The condensed nearest neighbour rule relies on the notion of a consistent subset. In it, the concept of consistency is based on the ability to perform accurate kNN classification. If the training set is not sufficient to perform accurate classification, then it is considered to be inconsistent. Specifically, a subset $\hat{E}$ of the training set S must be sufficient to perform correct kNN classification, with $k = 1$, of the remaining instances of S [Guo et al., 2008b].

Kubat *et al.*, in [Kubat and Matwin, 1997], propose one-sided sampling, which both cleans the border region via Tomek links and then removes distant points using the consistent nearest neighbour rule. Their motivation is to remove both the unsafe and uninteresting instances, leaving only the safe instances in the majority class training set.

More research into data properties is required to know which of the two biases is most appropriate or the degree with which to combine them. Moreover, when the imbalance is significant, a large number of majority instances may need to be removed, and the risk of lost information remains significant. Moreover, the reliance on a distance metric, typically the Euclidean distance, for the determination of the relative proximity of instances is of particular concern on manifold data and high-dimensional spaces in general. [Aggarwal et al., 2001], for example, discusses the fact that the notion of proximity in high-dimensional spaces may lack quantitative meaningfulness. Moreover, they demonstrate that the meaningfulness of the widely applied l_k norm, which includes the Euclidean distance (l_2 norm), is very sensitive to the choice of k on high-dimensional problems. In general, the Euclidean distance is an inaccurate measure of proximity on

high-dimensional data and manifolds known to be non-Euclidean on the macro-scale.

2.2.2 Cost-Sensitive Learning

Cost-sensitive learning aims to promote the induction of a classifier that minimizes the training error in a manner that accounts for the skewed misclassification costs that are often associated with problems of class imbalance. This is, in a sense, the most direct way to deal with the problem. The true misclassification costs are often unknown, however, rendering the application of cost-sensitive learning infeasible in some cases [Maloof et al., 1997, Maloof, 2003]. The set of cost-sensitive methods can be roughly divided into those that apply misclassification costs to the training data, those that utilize cost-sensitivity in conjunction with ensemble methods and the incorporation of cost-sensitivity into the learning algorithms themselves [He and Garcia, 2009].

Cost-sensitivity is applied to the training data as a form of bootstrap sampling. In [Sun et al., 2007], for example, various weight updating schemes are applied to AdaBoost in order to influence the induction process via the sampling of the training set. As previously stated, for this to work, it is required that the appropriate cost be known a-priori. Cost-sensitive fitting is intended to address the need for the actual costs to be known.

The fitting methods must be built into the algorithms themselves, such as decision trees and neural networks. Decision trees have been well studied in relation to cost sensitive learning [Breiman et al., 1984]. Cost-sensitive adjustments can be applied to the threshold, the split criteria at each node and to the pruning procedure. With respect to thresholds, it has been noted that there is a relationship between the class distribution and the placement of the threshold [Breiman et al., 1984]; however this is domain specific and can be hard to determine. [Maloof, 2003] suggest an alternative based on ROC analysis. Alternatively, the split criterion can be specified as an impurity function that is sensitive to skewed costs, such as Gini, Entropy or DMK [Drummond and Holte, 2000]. Finally, pruning has been found to have a negative impact on the learning of the minority class [Japkowicz and Stephen, 2002]; this is a natural result as pruning is intended to prevent overfitting noisy instances. In the case of class imbalance, noise and rare instances of value are indistinguishable. Thus, leaving out the pruning stage appears to be a wise decision. Doing so, however, has not proven beneficial [Japkowicz and Stephen, 2002]. Alternatively, [Elkan, 2001] suggests that better class probability estimates at the nodes allows for pruning to occur with more success.

When the minority class is small and/or the training set is linearly separable, in manner that is unrepresentative of the latent distribution, these methods will fail to have an effect [Wallace et al., 2011]. Particularly in cases of high-dimensional class imbalance, it is likely that the training set will be linearly separable due to sparsity, whereas the laten distribution is likely to be more complex. In these cases, we would like to shift the decision boundary; however we cannot achieve this with cost-adjustment. This is due to the fact that the decision boundary will only be affected if the cost-sensitivity causes some classifications to change. In such cases, the only means by which to shift the decision boundary is by undersampling the majority class [Wallace et al., 2011] or by adding more instances to the minority instances in a manner that expands the space towards the majority class. In domains of class imbalance, the additional instances must be synthesized. An additional benefit of resampling is that it they can be applied in conjunction with off-the-shelf learning algorithms, making for simple application. Due to the fact that sampling has been shown to produce similar [McCarthy et al., 2005] or better [Maloof, 2003] results than cost-based methods, it is often considered very appropriate. Based on this, the application of sampling methods in general, and oversampling specifically, can be seen to be preferential [Liu et al., 2007].

2.2.3 Oversampling

Random OverSampling produces a training set $S = \{E \cup S_{maj}\}$, where E is a set of minority training instances sampled with replacement. It rebalances the training distribution; however, it does not prevent the risk of overfitting in this case [Batista et al., 2004b]. In order to avoid overfitting, new instances must be added to the minority class. The practical limitations that are typical of class imbalance dictate that new instances of the minority class cannot be sampled, and thus, the learner must make do with the available information.

Synthetic data offers the possibility to augment the minority class with novel instances; this minimizes the risk of overfitting and, if representative instances of the minority class are generated, can lead to a much more robust classifier. An example of the use of synthetic data can be seen in the verification of the Comprehensive Test-Ban-Treaty [Stocki et al., 2010]. In this task, synthetic data was utilized to represent clandestine tests of nuclear weapons for the induction of binary learners for the verification of the treaty and to evaluate the performance of one-class classifiers in [Bellinger and Japkowicz,]. Artificially generated gamma-ray spectra with cobalt signatures were

synthesized by the Radiation Protection Bureau at Health Canada to aid in the evaluation of machine learning solutions for securing the Vancouver 2010 Olympics [Sharma et al., 2012b], and the attack class of the DARPA data set was synthesized to test the appropriateness of machine learning algorithms on the task of computer intrusion detection. In addition, handwritten characters were synthesized by rotating and skewing the training examples in [Ha and Bunke, 1997]. Although these methods have largely been effective in achieving their desired ends, they all utilize domain specific solutions that require considerable expertise and time. In general, however, we prefer a solution to the class imbalance problem that does not rely on domain simulations.

To be generally applicable, the synthesization method must not require that we recreate underlying generating processes explicitly. Rather, it is desirable that the synthetic instances can be generated based on the information that exists in the minority training set and any pre-existing knowledge of the data distribution or properties. An appropriate synthesization process should produce training data that affects a change in the induced classifier such that the system is better able to identify novel instances of the minority class with minimal affect on the majority class.

Chawla *et al.* presented Synthetic Minority Oversampling TEchnique (SMOTE) in [Chawla et al., 2002] as a general purpose synthesizer to overcome the limitation of random oversampling by generating a set of synthetic minority training instances on the edges connecting minority samples. It is predicted that generating new minority class instances, rather than replicating existing instances, will cause the classifier induction process to "carve out" a larger area of the feature space for the minority class. As a result, an improvement in the performance on the minority class is expected.

Explicitly, SMOTE augments the minority training set by interpolating points between nearest neighbours in S_{min}. For each $x_i \in S_{min}$, a synthetic point is created at a random point on the edge connecting x_i to a random instance x_j in its kNN set, $x_j \in kNN(x_i)$. Given $x_i,\ x_j$ and a random number $\delta = [0, 1]$, the synthetic point is calculated as

$$x_{new} = x_i + (x_j - x_i) \times \delta. \tag{2.1}$$

SMOTE has a few weaknesses that have been well documented in recent years. It has, for example, been shown to suffer from high variance and reinforcing noisy instances [Wang and Japkowicz, 2010, Stefanowski and Wilk, 2007, Batista et al., 2004c]. This results from the fact that it synthesizes new instances from a random nearest neighbour in a contextual vacuum, where the class distribution in the neighbourhood of these points is ignored. If there are noisy instances in the training set, the synthesized instances will

reinforce this noise, causing the induced classifier to generalize in directions unrepresentative of the latent distribution. A similar issue arises when SMOTE creates links between disjunct modes in the training distribution. As a thought experiment, imagine two subconcepts of the minority class separated by a concept of the majority class. In this case, SMOTE will synthesize new instances in the majority space along the edges connecting minority instances in the distinct subconcepts.

Another major issue is that due to the interpolation method, no synthetic points will be generated outside the convex-hull formed by the minority training set; in this respect, SMOTE under-generalizes on some problems. In scenarios where the minority training instances are linearly separable from the majority class, applying SMOTE will have no impact on the decision boundary. [Wallace et al., 2011] argue that this is a likely occurrence in high-dimensional problems.

To reduce the risk of over-generalization, the removal of Tomek Links or post processing with Wilsons Edited Nearest Neighbour rule has been proposed after the application of SMOTE [Batista et al., 2003, Batista et al., 2004a]. Other researchers have attempted to account for the context of the neighbourhood when synthesizing new instances. This is done, for example in [Han et al., 2005], where the nearest neighbour algorithm is applied to find minority instances on the border of the majority class to which SMOTE should be applied. The aim here is to populate the border region in order to combat the risk of under-generalization. Finally, He *et al.,* proposed that the number of synthetic instances generated for each "real" minority training instance can be decided dynamically via Adaptive Synthetic Sampling [He et al., 2008]. In particular, this method uses the portion of kNN for each minority instance x_i belonging to the majority class to decide how many synthetic instances to generate for x_i. In general, it uses the density of majority points around each minority instances to decide how many more minority instances should be generated. Thus, it puts more synthetic instances in regions close to many majority instances. i.e., on the border

Each of these methods relies on a straight line distance measure, typically in Euclidean-space. The use of the distance measure has been demonstrated to cause SMOTE to be less effective on high-dimensional domains than other imbalance classification methods [Blagus and Lusa, 2012]. We have noticed that it suffers from the contradictory problems of over-generalization and under-generalization. This, we argue, can be linked to high-dimensionality and the presence of the manifold property. In particular, SMOTE takes a localized perspective that can cause it to generalize in directions that are not associated with the latent manifold.

The relationship between local and global model building has been studied in the statistical inference literature and within the manifold learning domain itself [Hand and Vinciotti, 2003, Silva and Tenenbaum, 2002]. In line with the work of Hand and Vinciotti, by global we are referring to an inference process that aims to capture, and is accordingly selected based on, all aspects of the data distribution. This perspective is particularly appropriate when all regions of the data space are important since the global approach to model building can provide a more accurate representation of the underlying structure [Silva and Tenenbaum, 2002]. Alternatively, if we have some prior knowledge that a sub-region of the data is significantly more or less important, a local approach can be beneficial. SMOTE, however, is not applied in the context of explicit data properties; rather it tends to be applied in a knowledge vacuum as it relates to the properties of the target data. For an appropriate use of it, one must take the properties of the target data into consideration, and avoid its use on data that conforms to the manifold assumption.

Blagus and Lusa suggest that the negative impact of dimensionality can be combated by pre-processing with a feature selection method [Blagus and Lusa, 2012]; however, it is widely held by the broader machine learning community that manifold-based methods are the most effective means of addressing high-dimensional data that conforms to the manifold assumption [Mika et al., 1999, Roweis and Saul, 2000, Zhang and Zha, 2004, Hinton and Salakhutdinov, 2006]. This is, indeed, a much more direct solution, necessitating the development of our proposed framework, in contrast to the many solutions previously proposed that address the symptoms of SMOTE's behaviour.

Thus, the fundamental issue that we see in SMOTE is that it attempts to be too generic as it naively assumes a straight-line distance to data synthesization. As a result, the synthesization process lacks a perspective on the topology of the training distribution, and thus risks overpopulating regions that are already clustered with instances. Moreover, the applying straight-line approach to manifold data risks grouping distant points into the k-nearest neighbour sets and placing synthetic instances in regions of low density that may negatively impact the induced classifier. It is, indeed, this shortcoming of the SMOTE algorithm that the above adaptions aim to indirectly address. Their shortcoming, however, is in addressing only the symptoms of the problem, rather than identifying the root cause. One might envision that the use of a geodesic distance measure in the SMOTE algorithm would offer an effective solution. The lack of prior knowledge regarding the nature of the topological space in most domains, however, renders the application of such specialized distances measures impractical for addressing the weaknesses of SMOTE.

Our proposed method assumes that there is an embedded manifold structure, and uses the training data to infer the topological structure of the embedded manifold. As a result, the manifold-based framework produces a global representation of the data that is robust with respect to dimensionality. On appropriate datasets, our method directly minimizes the risk of overgeneralization away from the manifold. As a result, the synthetic data occupies regions of the data space that are most likely given the global topology of the training instances and the evidence that they collectively provide.

In spite of its limitations, SMOTE has been widely recognized for its ability to synthesize new instances for the minority training set. Together, these facts have motivated a burgeoning field of research into the generation of synthetic instances in scenarios of class imbalance. Lui *et al.*,, in [Liu et al., 2007], proposed a simple yet intuitive generative oversampling method based on an assumed parametric distribution with the parameters estimated from the minority training set. The authors claim, for example, that low-dimensional datasets can be modelled with a mixture of Gaussians [Melville and Mooney, 2004], and that text data is widely assumed to be modelled well by a mixture of multinomial distributions [McCallum and Nigam, 1998b]. For demonstrative purposes, the authors apply a multinomial model with Laplace smoothing on text data. They find this approach to work well on a variety of text datasets.

As a result of the Laplace smoothing, the proposed method is capable of synthesizing instances outside the convex-hull and can include words that are not present in the minority training set. The degree of generalization is controllable via the smoothing parameter, which is a beneficial feature when the minority training set is small and additional generalization over the minority class is beneficial. In comparison to random undersampling, they find their method to have lower variance.

The fundamental weakness of this generative method is the need for a suitable parametric distribution to represent the minority class. Such information is available for some domains, in which case the method has great potential. A large number of machine learning problems exist where the data is non-parametric; these include manifold data. Under these conditions a more general, non-parametric method is required. The work in this thesis proposes such a method.

Similarly, Gao *et al.,* proposed a method that is intended to improve upon SMOTE by generalizing beyond the convex-hull formed by the minority training set [Gao et al., 2012]. Unlike [Liu et al., 2007], this is done non-parametrically using parzen Window

with kernel density estimation. Specifically, the PDF $p(\hat{x})$ is estimated as:

$$\hat{p}(x) = \frac{1}{N_+} \sum_{i=1}^{N_+} \phi_\sigma(x - x_i) \tag{2.2}$$

where σ is a smoothing parameter and ϕ is a user-specified kernel function. For simplicity, a normal kernel scaled by a single sigma of the form

$$\phi_\sigma(x - x_i) = \frac{\sigma^{-m}}{(2\pi)^{m/2}} e^{-\frac{1}{2}\sigma^{-2}(x-x_i)^T(x-xi)} \tag{2.3}$$

can be applied. This implies that all dimensions of the feature spare are uncorrelated and have the same spread. Alternatively, utilizing a covariance matrix S offers the potential of a better estimate of the PDF

$$\hat{p}(x) = \frac{det(S)^{-1/2}}{N_+} \sum_{i=1}^{N_+} \phi_\sigma(S^{-1/2}(x - x_i)), \tag{2.4}$$

with the normal kernel function

$$\phi_\sigma(S^{-1/2(}x - x_i)) = \frac{\sigma^{-m}}{(2\pi)^{m/2}} e^{-\frac{1}{2}\sigma^{-2}(x-x_i)^T S^{-1}(x-x_i)}. \tag{2.5}$$

Oversampling is performed by randomly selecting n instances with replacement from the minority training set and using each of these as the key for generating a new instance. In particular, each sampled instance initiates a synthetic instance by utilizing the sampled instance as the mean and $\sigma^2 S$ as the covariance. Thus, the synthetic instance x_n resulting from the sampled instance x_0 is

$$x_n = x_0 + \sigma R \cdot \mathcal{N}(0, \sigma) \tag{2.6}$$

where R is the upper triangular matrix that is the Cholesky decomposition of S and $\mathcal{N}(0, \sigma)$ is an m-dimensional pseudo random number drawn from a zero-mean normal distribution.

On the target domain of tempered ductile iron classification and five UCI data sets, the authors find their method outperforms SMOTE in terms of the F-measure. Although, this method generates outside the convex-hull, it is similar to SMOTE in the sense that it performs a sort of local generation rather than generating synthetic instances based on a global perspective. Generation by applying kernel-based density around each training instance increases the likelihood of synthesizing instances outside the convex-hull; however, it also increases the risk of synthesizing instances in areas that do not belong to the minority class.

Finally, Fecker *et al.,* in [Fecker et al., 2013], propose a density induced oversampling method that takes advantage of the information in the majority class. This is said to enable the method to function in cases of extreme rarity where the existing methods fail. In particular, the authors propose to estimate a PDF of the majority class via a mixture of Gaussians and subsequently adapt the estimated PDF to include the minority training instances as well. In this way, they induce an adapted model of the joint minority and majority class training sets.

Algorithmically, the initial mixture of Gaussians is estimated via expectation maximization, which iteratively computes the mixture of Gaussians using the maximum likelihood. The mixture takes the following Gaussian form:

$$\mathcal{N}(x;\mu,\Sigma) = \frac{1}{(2\pi^{d/2}|\Sigma|^{d/2})} e^{-d/2(x-\mu)^T\Sigma^{-1}(x-\mu)}, \tag{2.7}$$

where μ is a d-dimensional mean vector, σ is a $d \times d$ covariance matrix and $|\Sigma|$ is the determinant of Σ. The proposed method assumes that the features of the data-space are independent, and thus, only the diagonal of Σ, written as $\sigma^2 := diag(\Sigma)$ is utilized. The resulting Gaussian mixture model takes the form:

$$\hat{p}(x|C_0) = \sum_{k=1}^{K} C_{o,k} \cdot p_k(x|C_0) = \sum_{k=1}^{K} C_{o,k} \cdot \mathcal{N}(x;\mu_{0,k},\sigma_{0,k}^2), \tag{2.8}$$

where x is the feature vector and K is the user-specified number of Gaussians and $C_{0,k}$ is a weighting factor.

The second step in the process is to adapt the model of the majority class to account for the minority class as well; Bayesian adaption is used for this. Bayesian adaption approximates a PDF from both the sparse minority class and the PDF estimated from the majority class in the first stage of the algorithm. Since maximum likelihood estimates of sparse data are often inaccurate, the authors argue that utilizing the prior knowledge in the majority class enables a more accurate estimate.

The adaption process works by probabilistically aligning each instances $x_1(m) : m = 1...M$ with the K Gaussian components

$$\hat{p}(k|x_1(m)) = \frac{c_{0,k}p_k(x_1(m)|C_0)}{\Sigma_{k=1}^{K} C_{0,k}p_k(x_1(m)|C_0)}. \tag{2.9}$$

The parameters of $\hat{p}(k|x_1(m))$ are combined with the coarse statistics calculated from the minority class, and the new weights of the components of K of the adapted PDF are obtained by:

$$\hat{C}_{1,k} = \alpha_k \cdot \frac{\Sigma_{m=1}^{M} P(k|x_1(m)) \cdot x_1(m)}{\Sigma_{k=1}^{K}\Sigma_{m=1}^{M} P(k|x_1(m))} + (1-\alpha_k) \cdot C_{0,k}. \tag{2.10}$$

The new mean takes the form:

$$\hat{\mu}_{1,k} = \alpha_k \cdot \frac{\Sigma_{m=1}^{M} P(k|x_1(m)) \cdot x_1(m)}{\Sigma_{m=1}^{M} P(k|x_1(m))} + (1 - \alpha_k) \cdot \mu_{0,k}, \tag{2.11}$$

and the variance is similarly calculated. The variable α_k, where

$$\alpha_k = \frac{\Sigma_{m=1}^{M} P(k|x_1(m))}{\Sigma_{m=1}^{M} P(k|x_1(m)) + r}, \tag{2.12}$$

controls the the influence of the majority class statistics versus the influence of the minority class statistic, where r is a fixed relevance factor controlling the influence of the minority training set on the adapted model. Finally, the adapted PDF, which includes the joint influence of both classes, is formulated as

$$\hat{p}(x|C_1) = \Sigma_{k=1}^{K} \hat{c}_{1,k} \cdot \mathcal{N}(x; \hat{\mu}_{1,k}, \hat{\sigma}_{1,k}^2). \tag{2.13}$$

As stated, the induced model represents the influence of both classes. Thus, an extra step is required to produce samples of the minority class. More specifically, synthetic samples are drawn from $\hat{p}(x|C_1)$, which may in fact belong to either class; thus, a filtering procedure must be applied to discard those instances generated from $\hat{p}(x|C_1)$ that are believed to belong to the majority class.

The authors propose and empirically test assignment based on Bayesian classification and the weighted nearest neighbour. They find that the Bayesian method tends to perform better with respect to the f-measure; however, the number of false positives increase in low probability regions of the majority space near the minority class. Alternatively, the distance based approach is more conservative in these areas. In general, the authors find that their method outperforms SMOTE and generative oversampling [Liu et al., 2007] in terms of the f-measure; this is particularly the case when the minority training set is very small. The authors argue that SMOTE requires a large number of minority training instances due to the fact that it does not synthesize instances outside the convex-hull.

In spite of the fact that their experiments suggest the proposed method has good potential, they note that it can be difficult to select an appropriate value for K. Moreover, the required ability to distinguish instances of the induced joint distribution from the majority class distribution is problematic. For small minority training sets, the difference between the two distributions is minimal. In addition, if the ability exists to distinguish them, it appears that the problem is solved prior to the induction of a binary classifier on the synthetically oversampled training set. Most importantly, like the others, this method

does not have the means of effectively handling data that conforms to the manifold assumption

2.3 Manifold Learning

It is widely held that the solution to many machine learning tasks involves the implicit discovery of the lower-dimensional manifold of the target domain [Chapelle et al., 2006]. This is related to the fact that the manifold space is believed to hold the natural representation of the data [Belkin and Niyogi, 2004] and that most machine learning algorithms suffer from the curse-of-dimensionality. The curse-of-dimensionality states that for increasingly higher dimensional problems, exponentially more training samples are needed [Bellman, 1961, Hughes, 1968, Clarkson, 1994]. Thus, finding the lower-dimensional manifold of the data is beneficial in terms of the number of training instances needed, and discovering the manifold-space implicitly simplifies the problem by revealing a better representation of the target data. This is intuitively beneficial to problems of class imbalance, particularly if the data conforms to the manifold assumption.

It is often the case that our knowledge of the data properties, either acquired from domain experts or from a general understanding of the features, can facilitate a reasonably good hypothesis regarding conformance to the manifold assumption. By analyzing the physical properties of a handwritten zero, for example, we see that it is fairly accurately represented by an ellipse. Because an ellipse is fully determined by the coordinates of its foci and the sum of the distance from the foci to any point, the ellipse can be represented by a five-dimensional manifold [Belkin and Niyogi, 2004]. A class of handwritten zeros is, indeed, more complex than an ellipse, so it will require more than five dimensions, but significantly fewer dimensions than the 28×28 feature-space of the grey-scale MNIST dataset, for example. In other domains, it is harder to identify conformance to the manifold assumption, however, it is a reasonable assumption for many high-dimensional domains where we can intuitively see that it has a complicated intrinsic structure that only occupies a small portion of the feature space [Belkin and Niyogi, 2004].

A significant amount of research has been devoted to the derivation of manifold learning methods. Interested readers are directed to the many high-quality surveys in the literature for a complete formalization of manifold learning [Huo et al., 2007, Izenman, 2011]. Many of the manifold learning methods that are described in these surveys, including Principle component analysis (PCA), kernel PCA, ISOMAP, local linear embedding and Laplacian Eigenmaps, have been applied in the machine learning literature to in-

fer the manifold-space in problems such as dimensionality reduction [Saul and Roweis, 2003, Roweis and Saul, 2000], feature selection [Xu et al., 2010], denoising [Mika et al., 1999], text classification [Zhang and Chen, 2005], image recognition [Weinberger and Saul, 2004], classification [Crawford and Ghosh, 2005, Tuzel et al., 2007, Yu et al., 2009] in general and unsupervised learning [Belkin and Niyogi, 2003, Belkin and Niyogi, 2004]; we find, however, that the existing methods of synthetically oversampling have been proposed and utilized without considering the manifold property [Chawla et al., 2002, Gao et al., 2012, Fecker et al., 2013, Shao et al., 2014]. Alternatively, they apply biases such as the placement of synthetic instances inside the convex-hull, generation according to a parametric assumption or the use of a kernel. None of these biases, however, are suitable when the data conforms to the manifold property and only a small number of training instances are available.

Like the learning algorithms that the current state-of-the-art in synthetic oversampling algorithms are intended to aid, the biases existing within these methods render their suitability to a particular problem dependent on the characteristics of the data. SMOTE, for example, implicitly assumes that the best representation of the minority class can be constructed from the convex-hull formed from the minority training instances in a Euclidean-space [Chawla et al., 2002].

The nearest neighbour bias in SMOTE fails based on the theory of manifolds, which states that Euclidean-space is only valid in the local neighbourhood of each point on the manifold. In scenarios of imbalanced classification, particularly high-dimensional class imbalance, the sparsity in the training set dictates the minority instances will be far apart; therefore, we cannot reasonably assume that the convex-hull will reside in a single, local neighbourhood. If the instances do not reside in a local neighbourhood of the manifold, then the use of the Euclidean distance as a measure of proximity is inaccurate. Indeed, the rarity and dimensionality of the minority class almost certainly dictates that the minority training instances will span large distances on the manifold, leading to poorer than necessary performance (a detailed description of why the various synthetic oversampling methods fail is provided in the next chapter).

Based on the above fact, an appropriate synthetic oversampling method must utilize the manifold representation. As we have shown above, however, none of the existing methods have attempted to do this. For this reason, we present the work of manifold learning as it has appeared in other areas of machine learning, and utilize this as a partial justification of our hypothesis that they can successfully be applied for modelling machine learning domains. Our work proposes a means of synthetically oversampling

from these models in order to mitigate the negative effects of class imbalance in domains that conform to the manifold assumption.

PCA is the standard method applied to perform this transformation. PCA is a linear mapping from the d-dimensional input space to a k-dimensional transformation space where $k < d$. The process involves calculating the eigenvectors $\mathbf{e}$ and eigenvalues λ from the co-variance matrix Σ. The $\mathbf{e}_{1...k}$ associated with the k smallest λ are considered most representative, and thus, form the transformed space [Pearson, 1901]. PCA can similarly be achieved by training an autoencoder via backpropagation with compression and linear activation in the hidden layer. This case, the k value is specified a-priori as the number of hidden units, where $k < d$. The principle components are thus obtained by discarding the output layer after training [Baldi and Hornik, 1989, Duda et al., 2001].

A strength of PCA in comparison to some other manifold learning methods, such as Local linear embedding, orthogonal locality preserving projections and ISOMAP, is that it does not rely on a local distance measure in the calculation of the manifold representation. Whilst this is not a issue in traditional manifold learning domains, it may be an issue from the perspective of synthetic oversampling where the minority training set is small. This is an aspect to be aware of when selecting the manifold learning method to utilize in our proposed framework; indeed, it is part our justification for using PCA and denoising autoencoders in our implementations.

PCA is often utilized as baseline system in manifold learning due to its efficiency and mathematical tractability. It was, for example, tested as the manifold mapping for image-based human age detection in [Guo et al., 2008a]. The authors found the manifold learning methods local linear embedding and orthogonal locality preserving projections to be superior for the target domain due to their abilities to incorporate class information.

In general, linearity is often seen as the major limitation of PCA. For most practical problems, the linear transformation is considered to be insufficient. As a result, a number of non-linear transformations have been proposed [Mika et al., 1999, Roweis and Saul, 2000, Zhang and Zha, 2004, Hinton and Salakhutdinov, 2006]. Non-linearity can be achieved in an autoencoder via appropriate regularization and non-linear activation at the hidden layer [Japkowicz, 2001, Vincent et al., 2010, Alain and Bengio, 2014]. Given the dearth of data in class imbalance, complex non-linear functions are not necessarily advisable. It is, nonetheless, beneficial to have such tools at our disposal when warranted. For this reason, we propose a framework that enables selection of a manifold learning method appropriate for the target domain rather than enforcing the use of a specific method.

Local linear embedding and ISOMAP are non-linear manifold learning methods that have successfully been applied in various machine learning contexts [Huo et al., 2007, Saul and Roweis, 2003, Guo et al., 2008a, Pan et al., 2009]. Both are non-linear and preserve the relative distance between points in the manifold space, whereas PCA does not. This is often considered to be preferable; particularly for dimension reduction and visualization. Provided the structure of the manifold is accurate, however, we believe that the preservation of the relative distances is less important from the perspective of synthetic oversampling. This is because we are not interesting in visualization, for example, where the relative position is of keen interest. Rather, we are interested in inferring the shape of the manifold from which to sample new training instances.

Local linear embedding is performed by first calculating the k-nearest neighbours of each instance $\mathbf{d}_i$ in the dataset $\mathbf{D}$ as $kNN(D)$. The index of each $kNN(\mathbf{d}_i)$ in $\mathbf{D}$ is recorded in the list N_i. The optimal local convex combination is then found for kNN set of $\mathbf{d}_i$ as:

$$\epsilon\big(\mathcal{W}\big) = \sum_i \big|\mathbf{d}_i - \sum_{j \in N_i} \mathcal{W}_{ij}\mathbf{d}_j\big|, \tag{2.14}$$

where $\sum_j \mathcal{W}_{ij} = 1$. Finally, the project set $\mathbf{Y}$ of $\mathbf{D}$ is calculated by minimizing the objective function:

$$\phi\big(\mathbf{Y}\big) = \sum_i \big|\mathbf{y}_i - \sum_{j \in N_i} \mathcal{W}_{ij}\mathbf{y}_j\big|. \tag{2.15}$$

ISOMAP is an extension of multidimensional scaling that replaces the Euclidean distances with an alternate distance measure. ISOMAP commences by finding the set of instances in the neighbourhood of $\mathbf{d_i}$ in the dataset $\mathbf{D}$ based on the kNN set or a local radius. Similar to local linear embedding, the indices of the instances in the neighbourhood of $\mathbf{d}_i$ are placed in the list N_i. A graph is then created with edges connecting each $\mathbf{d_i}$ and $\mathbf{d_j}$ where $j \in N_i$ or $i \in N_j$. The distance between each $\mathbf{d_i}$ and $\mathbf{d_j}$ in the graph is specified as the sum of the arc length of the shortest chain connecting $\mathbf{d_i}$ and $\mathbf{d_j}$. From this, the lower dimensional projection is calculated according to multidimensional scaling.

The above formalizations of local linear embedding and ISOMAP serve to demonstrate their reliance on distance-based derivations of the implicit manifold. As we previously indicated, we do not want to rule out any manifold learning methods outright; however, we do want to emphasize the possible risks of such mechanisms when the training set is small.

Non-linearity can also be achieved with kernel PCA [Mika et al., 1999]. The primary

algorithmic difference in Kernel PCA and PCA is that kernel PCA computes the covariance matrix $\mathbf{\Sigma}$ in the kernel-space ϕ. As in support vector machines, however, much of the computational complexity is removed by applying the *kernel trick*, which enables the operation to take place in an implicitly infinite space, but the computation to be made in the lower-dimensional space by simply computing the inner products [Vapnik, 1995].

From the perspective of synthetic oversampling, the primary deficiency of many manifold learning methods is that they require us to operate in the lower dimensional manifold space. This is, in fact, the desired outcome of manifold learning in domains such as dimensionality reduction; for synthetic oversampling, however, we must draw samples from the manifold in the manifold space and maintain the ability to map them back to the feature space for classifier induction. In this context, PCA and autoencoders offer ideal manifold representations because of the intuitiveness of the sampling mechanisms that are implicit in their formulation. We find the denoising autoencoder to be particularly suitable.

What we, in addition to others, have recognized in the autoencoder is the propensity of a well trained autoencoder to map input to the manifold in both the reduced space and feature space [Rifai et al., 2012, Alain and Bengio, 2014]. Our specific contribution is that we have recognized this property for its helpfulness in the context of synthetic oversampling. Moreover, we have generalized this notion of modeling and sampling the latent manifold to a framework in which other manifold learning algorithms can be applied. For the time being, however, we focus on autoencoders sampling because autoencoders have previously been qualitatively evaluated in the literature based on the samples that they can generate.

An autoencoder is an artificial neural network formed of a input layer, hidden layer and output layer [McClelland and Rumelhart, 1986]. In the traditional setting, the network forms a bottleneck in which the input and output layers have one unit for each feature in the input feature vector and the hidden layer has strictly fewer units than the number of features. This is depicted in Figure 2.1. The autoencoder is well-known for its applications in supervised learning where the network is trained autoassociatively to reconstruct the input at the output layer after having passed it through the bottleneck [Anderson et al., 1977, Kohonen, 1977]. During the training process, the reconstruction error is backpropagated down the network to update the weights, thereby improving the reconstructive capabilities over multiple epochs [Rumelhart et al., 1986, Rumelhart et al., 1995].

Regularized autoencoders, such as contractive auotencoders and denoising autoen-

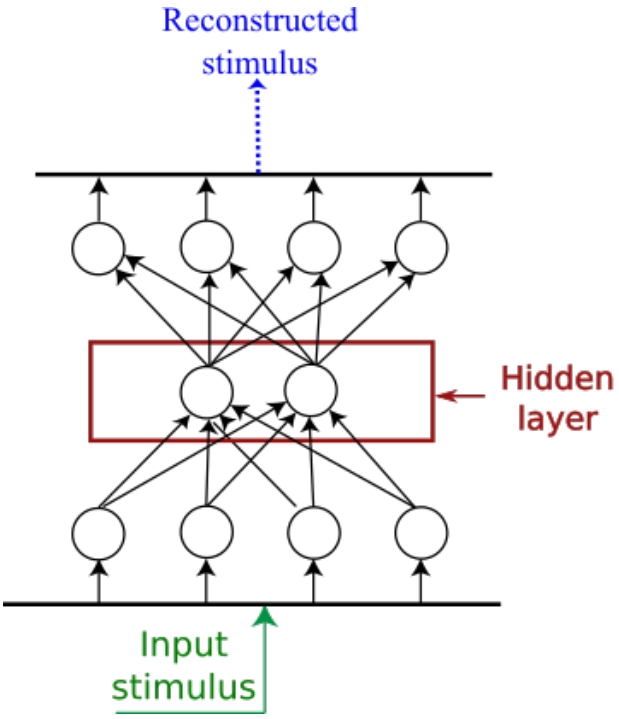


Figure 2.1: General form of an autoencoder.

coders, have been studied in relation to their inductive abilities on domains involving lower-dimensional manifolds [Rifai et al., 2012, Alain and Bengio, 2014]. In [Rifai et al., 2012], it is demonstrated that these methods capture the local manifold structure through the leading singular vectors of the Jacobian. In [Alain and Bengio, 2014], Alain and Bengio articulate the mechanics of denoising autoencoders as requiring that the reconstruction function be as simple as possible, and examples that are neighbours on the high-density manifold be represented differently. Thus, accurate reconstruction of points on the manifold is made possible, whilst the reconstruction error quickly rises for examples orthogonal to the manifold. This is specifically the behaviour that we require of a modelling process for the GRS datasets and other datasets that conform to the manifold assumption. This enables us to generate samples by mapping sample initiation points to the manifold via a trained denoising autoencoders.

Ranzato *et al.*,, in [Ranzato et al., 2008], proposed the first probabilistic interpretation of regularized autoencoders. This lead to further analysis of how to form a probabilistic understanding of autoencoders and, indeed, considerations on how to draw samples from the induced model [Vincent et al., 2010]. Based on the observation that the Jacobian matrix of derivatives of the encoding function provided estimates of the local density, a

sampling procedure based on encoding, decoding and the addition of noise was initially proposed [Rifai et al., 2012]. Alternatively, Bengio describes a sampling method based on Langevin and Metropolis-Hasting MCMC [Alain and Bengio, 2014].

Visual analysis of the proposed methods on artificial domains, handwritten digits and facial recognition tasks have been utilized to demonstrate the effectiveness of denoising autoencoders learning, and of the sampling procedures themselves. Due to the fact that the sampling methods take pseudo random walks down the learnt manifold, long walks are required to produce samples that cover the entire manifold. This increases the risk of getting stuck in high density regions. For a fast and efficient alternative to this, we utilize the ability of a well trained denoising autoencoders to map random sample instances sampled from a non-target distribution to the manifold for sampling. Furthermore, we propose model selection based on a reconstruction error ratio in order to ensure that samples can be mapped from all regions of the feature space onto the manifold.

Whilst we find that the denoising autoencoder offers a simple and efficient sampling method for synthetically oversampling datasets conforming to the manifold assumption, we recognize that each manifold learning method incorporates a specific bias and assumptions that render it more or less suitable for individual domains. Indeed, we see many benefits in the simple PCA method and demonstrate a novel means of synthetically oversampling a PCA model in the Chapter 4. In addition, a means of generating samples from a local linear embedding has also appeared in the literature [Juanjuan et al., 2007]. It is for these pragmatic reasons that we propose a framework for synthetically oversampling the manifold in the pending chapters rather than suggesting a single, limiting option.

2.4 Conclusion

Class imbalance is widely held to be one of the outstanding challenges in modern machine learning. As a result, researchers have given the quest for solutions to the induction of binary classifiers on imbalanced training sets a considerable amount of attention. The primary focus can be dichotomized into sampling-based methods and cost-sensitive solutions. As shown in this work, whilst the state-of-the-art has led to an improved ability to handle class imbalance, there still exists a number domain attributes upon which the existing methods underperform. In particular, undersampling methods often remove important instances, and even when advanced undersampling methods are applied, they

can fail due to the necessity to remove a significant number of majority class instances on heavily skewed learning task, or due to the application of an incorrect bias. Alternatively, cost-sensitive methods require domain knowledge regarding misclassification costs that is often unavailable and necessitates the modification of existing learning algorithms.

The above two facts motivate the use of oversampling; however, it is widely held that standard random oversampling does not add new information, and thus often leads to overfitting. To address this, researchers have recently proposed synthetic oversampling as a way to avoid overfitting the minority class and to limit the need to heavily under-sample the majority class. Indeed, synthetic oversampling offers the potential to generate new instances of the minority class for use in classifier induction, leading to a more robust classifier of the minority class; the alternative methods for handling class imbalance cannot provide this.

Although the field of research into synthetic oversampling has begun to develop, it is still very much in its infancy. Only recently have simple parametric and non-parametric methods been proposed, and still important questions such as where to synthesize minority instances and how to model the training set for particular domains and particular degrees of rarity have not been thoroughly considered. Moreover, aside for the work of [Fecker et al., 2013], no work exists that directly considers synthetic sampling in cases of absolute rarity.

In this thesis, we consider the pertinent question of synthetically oversampling data that is often high-dimensional and conforms to the manifold assumption. To date, no research has directly addressed the highly valuable question of synthetically oversampling the manifold. In this chapter, we have outlined various manifold learning methods that might be used to achieve this based on their successful applications in other areas of machine learning, and suggested some key criteria for the algorithm with respect to synthetic oversampling.

In the subsequent chapters, we utilize this information to understand why the existing methods fail on manifold data and we propose a framework for synthetically oversampling data domains that conform to the manifold assumption; these include many high profile high-dimensional domains such as image, text and scientific data. Synthetic oversampling in this way performs well on high-dimensional data, where distance-based methods such as SMOTE fail. Moreover, when suitable, the framework allows for the induction of a non-parametric manifold that captures the key variations in the training data. From a data generation perspective, this means that unlike the discussed alternatives, the synthetic instances are generalized in important directions whilst the variation in other

directions is minimized. As a result, the synthetic set is more likely to only encroach on the majority class in places that are likely to improve classification of the minority class, whereas SMOTE and other existing methods, which generalize isotropically, cause the synthetic set to spread in unhelpful, and perhaps, harmful directions.

Chapter 3

Synthetic Oversampling and the Effect of the Manifold

3.1 Introduction

An aspect of this thesis is to promote the no-free-lunch theorem as it stands in the context of synthetic oversampling. We note that in many cases machine learning experts can utilize domain expertise to design learning algorithms that are more appropriate for the specific properties of the data to which they are to be applied.

It is, indeed, recognized by the manifold learning community that it is often necessary to identify conformance to the manifold assumption [Chapelle et al., 2006] in order to obtain good results in machine learning domains, such as those involving images, text, scientific data, environmental data and machine sensor data, *etc.* Despite its prominence in many areas of machine learning, the manifold assumption has yet to be considered within the subfield of class imbalance. In this chapter, we demonstrate that the methods considered to be the state-of-the-art in synthetic oversampling were designed without any consideration for the manifold. Moreover, we demonstrate that the performance gains offered by these methods degrade when the target domain conforms to the manifold assumption. This chapter servers as the jumping off point for the subsequent chapter in which we propose a framework for synthetically oversampling the manifold.

The manifold assumption is characterized by a data-space that is often high-dimensional, and in which the data probability density resides in a lower-dimensional manifold. In these scenarios the probability density can be said to live in a lower-dimensional space. Identifying conformance with the manifold property and applying an appropriate method

is often key to good performance on domains of this nature. As a result, a significant amount of research has been devoted to the derivation of methods capable of inferring the manifold-space from training sets [Mika et al., 1999, Roweis and Saul, 2000, Zhang and Zha, 2004, Hinton and Salakhutdinov, 2006].

This chapter commences by studying the impact of the manifold on "off-the-shelf" classifiers on imbalanced classification problems. In particular, we examine the degree to which classifier performance degrades as a result of the manifold and consider this to be an acceptable upper bound on degradation after synthetic oversampling. Specifically, we argue that the utilization of synthetic oversampling methods should not lead to more degradation than that incurred by the baseline classifier alone.

The result of our experiments confirm our hypothesis that the existing synthetic oversampling methods suffer when the manifold property exists. In particular, whilst the overall performance generally increases as a result of synthetic oversampling, the rate of degradation incurred when performing classifier induction after synthetic oversampling is generally worse than that of the baseline. This indicates that when the manifold assumption holds, manifold-appropriate methods should be applied in order to maintain the gains of synthetic oversampling.

3.2 Motivation

Synthetic oversampling methods are required to make large generative leaps from small training sets on real-world imbalanced classification tasks. Thus, in order to maximize the likelihood of synthesizing instances that are beneficial for classifier induction, it is of absolute importance to understand the relationship between synthetic oversampling methods and the data to which they are applied.

The metamorphosis of machine learning as an isolated discipline to data science means we are no longer left to base our solutions solely on the toolbox of advanced mathematics. Data science involves careful consideration of the data, and machine learning experts working in collaboration with domain experts. This facilitates the utilization of the best practices from both fields and promotes the effective design of intelligent learning system. To ignore the properties of the data, when they are known, is to place an unnecessarily low upper bound on performance.

In this chapter we are motivated to understand the relationship between the existing synthetic oversampling methods and the data that they are applied to. More specifically, we are motivated to understand the implications of ignoring the physical properties of the

data and applying generic methods. In our particular scope, we focus our attention on the relationship between data that conforms to the manifold property and the effectiveness of methods considered to be of the state-of-the-art in synthetic oversampling. In particular, we test the hypothesis that the typical methods of synthetic oversampling degrade with the presence of the manifold property in data.

3.3 Contribution

The primary contributions of this chapter are in the articulation of the need to consider the relationship between the data synthesization method and the properties of data to which it is to be applied, and the demonstration that the existing methods degrade on data that conforms to the manifold assumption.

3.4 Applied Algorithms

This section outlines what we consider to be fundamental methods of synthetically oversampling the minority training data for problems of class imbalance. In addition, we highlight the binary classification algorithms that are applied in the experiments to follow. A general understanding of the latter is important because the synthetic oversampling methods affect them differently depending on their algorithmic form.

3.4.1 Synthetic Oversampling Algorithms

Synthetic oversampling methods utilize a generative bias plus the minority training instances to model[1] the minority class and generate additional training instances for use during classifier induction. The authors of [Rifai et al., 2012] accurately articulate the objective of modeling from a finite training set as one of deciding upon how to redistribute the probability mass associated with the training instances.

The simplest means by which to redistribute the probability mass is to assume a parametric form, such as a Gaussian or a mixture of Gaussians, estimate the parameters from the training data and utilize the assumed distribution along with the estimated parameters to generate synthetic instances [Fecker et al., 2013]. Under certain conditions this approach is a very practical solution. Low dimensional datasets, for example, can be

[1]We use the term model loosely since SMOTE does not physically model data, but simply applies its bias to the training data.

modelled with a mixture of Gaussians [Melville and Mooney, 2004] and the multinomial distributions has been found to be effective when applied to text data [McCallum and Nigam, 1998a, Liu et al., 2007]. Making parametric assumptions, however, can have a negative impact on classifier induction if it is incorrect or if the estimated parameters are erroneous. This is a significant risk because the training sets are small and a parametric form may not exist. As a result, non-parametric methods are typically preferred.

Parzen-window offers a non-parametric alternative that involves estimating the density based on the number of training instances residing within each of the hypercubes in the set used to partition the space. A hypercube that has many instances within its boundaries is considered to represent a region of greater density. The length of the sides of the hypercubes are user-specified and allow the user to adjust the granularity of the non-parametric estimate. In theory, small hypercubes will lead to a more accurate representation; however, this is computationally very expensive and would produce a large number of empty hypercubes on high-dimensional training sets with few examples. In addition, a kernel can be applied to smooth the density transition between hypercubes. Kernel density estimation suffers considerably from the curse-of-dimensionality and is inappropriate for high-dimensional domains with rarity. This results from the fact that a large majority of hypercubes would have no training examples.

The most prominent non-parametric form of synthetically oversampling, and indeed, the first proposed in the literature for class imbalance, is SMOTE [Chawla et al., 2002]. Philosophically, it starts from the premise that under some circumstances, additional training instances are needed for the minority training set, and simple replications (random oversampling) are insufficient. In order to achieve this, SMOTE applies a bias that assumes the best place to insert probability mass is between nearest neighbours in the minority set; collectively, these form the convex-hull.

Algorithmically, SMOTE finds the k-nearest neighbours of each instance $\mathbf{x}_i$, and generates a synthetic point between $\mathbf{x}_i$ and a randomly selected instance $\mathbf{x}_j$ in the nearest neighbour set. Formally, the new point $\mathbf{x}_{new}$ is generated as:

$$\mathbf{x}_{new} = \mathbf{x}_i + (\mathbf{x}_j - \mathbf{x}_i) \times \delta \tag{3.1}$$

where the uniform random variable $\delta \in [0, 1]$ causes $\mathbf{x}_{new}$ to be placed at a random distance between $\mathbf{x}_i$ and $\mathbf{x}_j$.

The primary issues with SMOTE are that, depending on the properties of the data, it can risk either under-generalizing or over-generalizing the minority space. Specifically, the SMOTE process leads to the added density residing entirely inside the convex-hull

formed by the minority training instances [He and Garcia, 2009]. In addition, when the minority set is small and the dimensionality of the learning task is high, it is likely that there will be a large distance between nearest neighbours. With this comes a significant risk of synthesizing instances in areas of the feature space that should not be populated by the minority class. This spread is the motivation for expensive post-hoc cleaning processes, such as the removal of Tomek links [Tomek, 1976].

SMOTE+Tomek commences with the typical application of SMOTE to synthesize instances of the minority class. This is followed by a cleaning phase that finds the nearest neighbours of each synthetic instance in the combined minority/majority training set and removes those instances with neighbours belonging to the alternate class. Alternatively, borderline SMOTE first applies the kNN algorithm to find the set $\mathcal{B}$ of minority training instances that are on the border with the majority class and then applies the SMOTE algorithm to the set $\mathcal{B}$. This is utilized with the objective of generating synthetic instances in the minority space close to the majority class. In addition, it performs post-hoc cleanning to remove instances it believes to be noise[2]. We have found that borderline SMOTE is ineffective in cases of absolute imbalance.

3.4.2 Classification Algorithms

This section provides a brief overview of the binary classification algorithms utilized in these experiments. Our focus is primarily on the relationship between these algorithms and data with latent manifolds. We direct readers that desire further details regarding the algorithms to [Mitchell, 1997, Duda et al., 2001]. Our experiments utilized the Weka implementations of these classifiers [Hall et al., 2009].

Naïve Bayes is a probabilistic classifier based on Bayes theory. The unique aspect of naïve Bayes is that it naïvely assumes the data dimensions to be independent. The classifier utilizes the maximum posterior decision rule to perform classification based on the following equation:

$$\hat{y} = \arg \max_{k \in \{0,1\}} P(C_k) \prod_{i=1}^{n} p(x_i | C_k), \tag{3.2}$$

which is the product of the component densities. A result of taking the product is that components that are not helpful in distinguishing between classes are implicitly ignored. This can reduce the impact of the manifold in some cases. In other cases, however, it

[2]Each of these methods is discussed in greater detail in the literature review chapter, Chapter 2

is necessary to account for the dependence between features in order to produce good classification results. naïve Bayes is unable to achieve this.

The C4.5 decision tree progressively builds a decision tree in a top down fashion that attempts to select the attributes that best dichotomize the data for the top of the tree. The building process stops when the training data is sufficiently classified. In this manner, C4.5 marginalizes data attributes that are minimally beneficial for classification. Like naïve Bayes, however, it treats the features independently, which affects its ability to account for the dependencies between features.

The multi-layer perceptron (MLP) classifier is an artificial neural network in which the size of the input layers is equal to the dimensionality of the data, the size of the hidden layer is user-specified, but is typically smaller that the input layer to reduce noise, and the output layer has a single unit that specifies the classification. The network is fully connected from the input layer to the hidden layer and from the hidden layer to the output layer [McCulloch and Pitts, 1943, Hebb, 1949]. Learning involves the use of back-propagation and gradient descent to evolve the weights that connect the units from their initial random value to values that facilitate accurate classification. Through this process the network has the potential to learn representations that are more suitable for complex data than those learnt by NB and C4.5. Moreover, it captures dependence relationships between features; however, like all other learning algorithms that conduct a search through the hypothesis space, convergence to the global minimum is not guaranteed.

Support vector machines (SVM) are amongst the most widely applied and cited methods in the modern machine learning literature. They are arguably considered to be one of the best general purpose classification algorithms [Vapnik, 1995, Schölkopf and Burges, 1999]. When it is applied with the popular RBF kernel, the training set is mapped to a theoretically infinite kernel-space in which a linear separator can be, in theory, perfectly applied. In practice, however, slack variables are available to allow for class overlap in the kernel-space. This embedding in the higer-dimensional Euclidean-space does not necessarily provide a natural separating direction for manifold data. In particular, it is intractable to compute the explicit separating hyperplane, which is the foundation of many Euclidean-based classifiers, when generalized for an arbitrary manifold [Sen et al., 2008]. As a result of their Euclidean foundation, the standard SVM kernels are unable to account for the intrinsic geometric structure of a manifold [Zhang and Chen, 2005].

Finally, k-nearest neighbour (kNN) is the most susceptible to degradation on manifold data as it is commonly applied with the Euclidean distance (L_2). Specifically, kNN

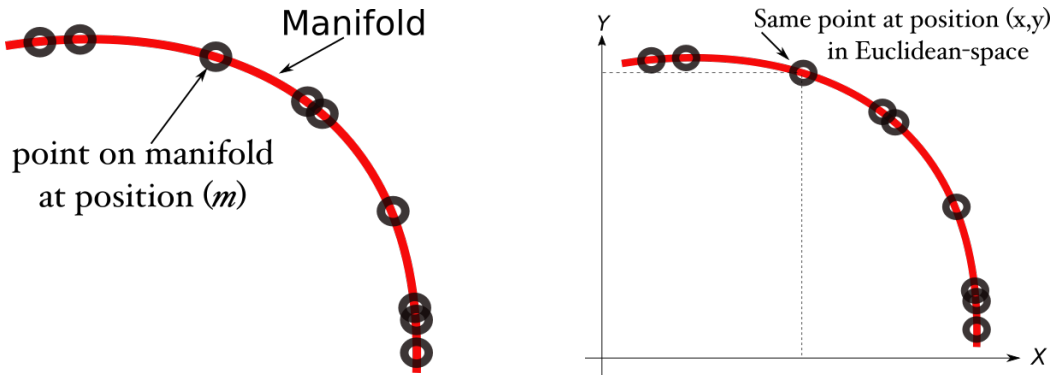


Figure 3.1: The subfigure on the left demonstrates a one-dimensional manifold in its manifold-space and the subfigure on the right places the one-dimensional manifold in two-dimensional Euclidean-space.

assigns a class label to a given instance, x, based on the class frequency of its k nearest neighbours. The Euclidean distance is a poor measure on manifold data and on high-dimensional data in general. In addition, it gives equal weight to each feature in spite of their relative importance. However, the generalized L_k norm with smaller k values has been shown to be more robust on manifold data [Aggarwal et al., 2001].

3.5 Problem Overview

Synthetic oversampling has contributed greatly to solving the problem of class imbalance. Nonetheless, the existing methods of synthetic oversampling have been proposed and utilized without considering the properties of the data to which they are applied [Chawla et al., 2002, Gao et al., 2012, Fecker et al., 2013, Shao et al., 2014]. Like the learning algorithms that they are intended to aid, the biases existing within these methods render their suitability to a particular problem dependent on the characteristics of the data.

Our motivation in this thesis is in relating synthetic oversampling methods to the manifold property. We commence this overview of the associated problem with a brief discussion of the manifold assumption. This is followed by an analysis of the impact of such data on the existing synthetic oversampling methods.

3.5.1 The Manifold Property

This section provides a brief overview of manifolds. Readers that are interested in further detail are directed to the following references [Ma and Fu, 2011, Izenman, 2012, LeCun et al., 2006]. Formally, a manifold is a topological space that resembles a Euclidean-space near each point. Each point of an m-dimensional manifold has a local neighbourhood that is homomorphic to an n-dimensional Euclidean-space. As we show, this has significant implications on how best to model and synthesize instances from data with the manifold property.

A common and effective analogy for understanding manifolds from a machine learning perspective is to think of sticky rice on a plate. Looking down on the rice, we can specify the location of each grain using a two-dimensional coordinate system on the plate. If we raise the plate off the table and tilt it, we would typically specify the location of each grain in the three-dimensional space represented by the room. Nonetheless, nothing has changed for the rice with respect to the plate. Therefore, we could still use the plate-based, two-dimensional coordinate system for the grains if we knew the orientation of the plate in the three-dimensional space. As has been found in the broader realm of machine learning, the latter is often the most effective means of representing data.

Manifold learning methods, such as PCA and autoencoders, learn the orientation of the manifold and specify the position of the samples on the manifold. The rotation matrix of PCA, for example, describes the mapping from the higher-dimensional space to the manifold-space (from the three-dimensional position in the room to the the two dimensional position on the plate, for example), and the eigenvectors represent the manifold coordinate system. For data that conforms to the manifold assumption, these methods offer a better, and lower dimensional, representation of the data. This has been shown to produce improved results for many machine learning applications.

From the perspective of synthetic oversampling, the lower-dimensional manifold space reduces the impact of the curse-of-dimensionality, which is of significant importance given the dearth of data. Even more importantly, this allows for the generation to be performed in the manifold-space, which is a more concise representation focused on the probability density of the data, thereby reducing the risk of synthesizing instances in low probability regions of the feature space.

Figure 3.1 illustrates a one-dimensional manifold in red with samples from the manifold appearing as black circles. Just as we could specify each grain of rice based on two coordinates from the two-dimensional surface of the plate (manifold), we can specify

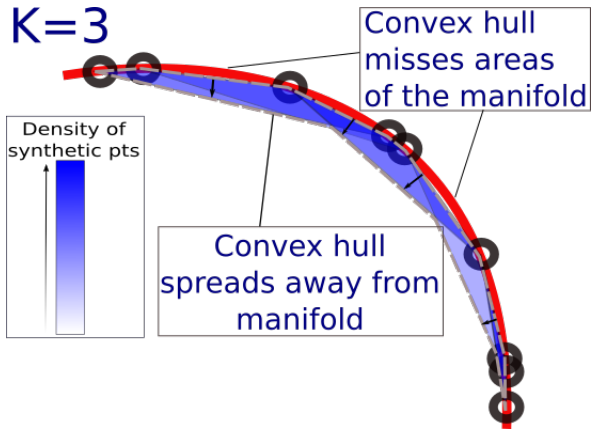


Figure 3.2: Demonstration of SMOTE-based synthesization of a manifold.

each instance here according to a single value relating its position on the manifold. Once again like the sticky rice, we can also imagine the points on the manifold in this figure as floating in a higher dimensional space. This is depicted in the right subfigure.

In the graphic on the right, each instance is specified based in its (x, y) values in the Euclidean-space. It is in this higher-dimensional space that the conventional synthetic oversampling methods perform their generation. In the subsequent section, we utilize this example of the one-dimensional manifold embedded in the two-dimensional Euclidean-space to demonstrate the risks associated with the existing synthetic oversampling methods when applied to data that conforms to the manifold property.

3.5.2 The Manifold and Synthetic Oversampling

This section demonstrates the impact of performing synthetic oversampling via non-manifold-based methods on data that conforms to the manifold-assumptions. The key points are that 9n each case, the synthetic sampling method:

- overgeneralizes away from the manifold; and
- under-generalizes along the manifold.

The following illustrations of the weaknesses of the conventional methods utilize the one-dimensional manifold embedded in the two-dimensional Euclidean-space that is depicted in Figure 3.1.

Figure 3.2 demonstrates the overgeneralization away from the manifold resulting from synthetic oversampling based on a kNN process of SMOTE. According to the mathematical properties of a manifold, kNN-based synthesization is only appropriate when

the training set is large enough to ensure that the set S_i of kNN of x_i, $S_i = kNN(x_i)$, resides in the close geometric neighbourhood of x_i. Under these conditions, an instance s'_i synthesized between x_i and a random kNN $s_i \in S_i$ will remain on the manifold. For increasingly small training sets, however, the distance between x_i and s_i becomes large, and thus extends outside the geometric neighbourhood of x_i. When this occurs, the synthetic instance s'_i can no longer be expected to reside on the manifold. This is precisely the case depicted in Figure 3.2.

The shaded area in this figure represents the convex-hull formed by the training instances (black circles). It is within this convex-hull that SMOTE synthesizes new instances for the training set. Because the convex-hull is formed by connecting distant instances on the manifold, it has a great risk of covering regions off the manifold. The black arrows in the figure serve to emphasize this risk on our exemplary problem. Smaller k-values may reduce the distance between the selected nearest neighbours, however, this is also likely to reduce the spread of the synthesized instances along the manifold. Indeed, rather than trying to balance the risk, it is more appropriate to apply a method suitable for manifold data.

Whilst SMOTE and its derivatives are by far the most prominent forms of synthetic oversampling in the machine learning literature, a small number of alternatives have more recently been proposed that utilize parametric assumptions, such as Gaussian or a mixture of Gaussians estimates [Liu et al., 2007, Fecker et al., 2013], and non-parametric kernel-based methods [Gao et al., 2012]. On manifold data, however, these methods present even less ideal alternatives than SMOTE.

The range of parametric assumptions, including Gaussian, multinomial, poisson, *et al.*, clearly do not hold when it comes to manifolds. On some machine learning tasks, parametric assumptions are applied somewhat blindly irrespective of the fact that they may not hold. This approach is problematic from the perspective of synthetic oversampling data that conforms to the manifold assumption because it leads to the spread of synthetic instances into spaces that are not part of the latent manifold. This is depicted for Gaussian generation over a manifold in Figure 3.3. Here, modelling the manifold in the two-dimensional space with the Gaussian distribution causes the centre of the density of the synthetic instances to be placed completely off the manifold. From a machine learning perspective, this causes the degradation of the performance of the induced classifier when the unrepresentative instances cause an erroneous shift in the decision boundary.

The non-parametric alternative is based on kernel density estimation. Here, the

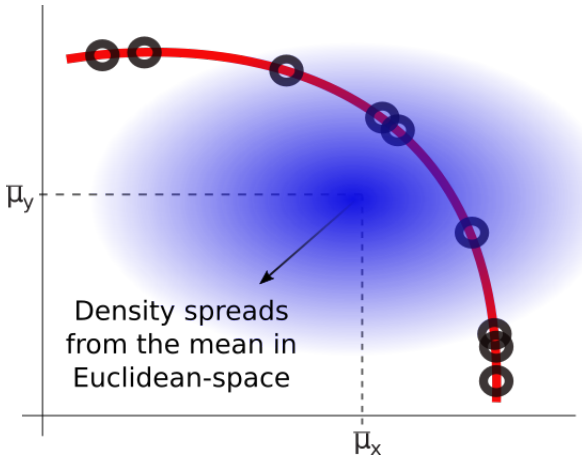


Figure 3.3: Demonstration of a Gaussian synthesization of a manifold.

feature space is divided into equal sized cuboids, the density is estimated in each cuboid from the number of training samples found within it. A kernel function, such as the RBF kernel utilized in [Gao et al., 2012], is then applied to smooth the density estimation across the cuboid boundaries. This is depicted over our hypothetical manifold in Figure 3.4.

Both under-generalization along the manifold and over-generalization away from the manifold are issues for kernel-based methods when applied to small training sets and high-dimensional data. Two things are clear from the graphic: *a)* many cells with the manifold passing through them are void of training instances, and thus instances are unlikely to be synthesized there, and *b)* individual cells cover both high density regions on the manifold and very low density regions off the manifold. The high density region passing through the cell increases the probability density of the entire cell, and the neighbouring cells, after kernel smoothing is applied. Thus, there is an isomorphic spread away from the manifold into areas of low probability [Rifai et al., 2012]. Therefore, cells along the manifold but far from training instances will receive low probability densities, and cells that are orthogonal to the manifold but near training instances on the manifold will receive higher density.

Although SMOTE clearly suffers from the same problems as these other algorithms, it does so to a lesser degree. Therefore, in the experimental evaluations SMOTE represents our best competitor and receives the majority of our attention in the subsequent chapter.

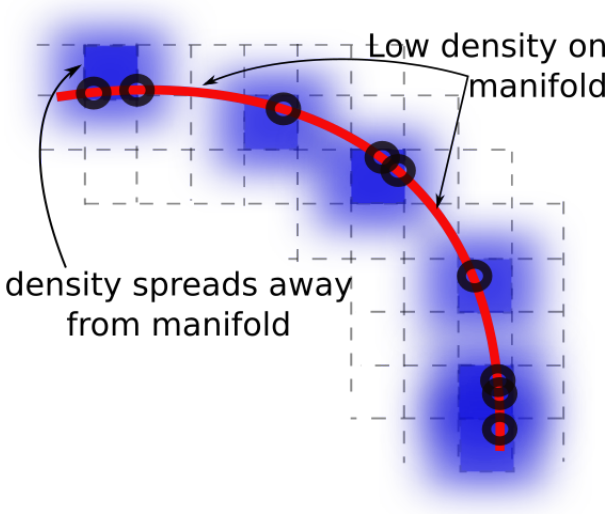


Figure 3.4: Demonstration of a kernel-based synthesization of a manifold.

3.6 Experimental Setup

The experimental methodology in this chapter is designed to test our hypothesis that the existing synthetic oversampling methods underperform on data that conforms to the manifold property. In order to test this hypothesis, we set up controlled experiments in which we can examine how the methods perform on manifold and non-manifold data.

Our initial approach to test the hypothesis was to utilize a large number of UCI datasets, which we would evaluate in terms of their conformance to the manifold assumption. Two issues exist with this method; there is no widely accepted generic mathematical test for the degree of conformance to the manifold assumption, and this methodology does not control for the many other factors that make classification difficult.

An effective alternative to the above is to synthesize a set of new datasets from existing UCI datasets that increasingly conform to the manifold property. The benefits of this are that;

- We start with existing, well-known benchmark datasets
- We control the degree to which conformance to the manifold property is increased for the data
- All other complicating factors of the data remain constant; and
- We can evaluate the affect of the manifold based on the change in performance.

Using this methodology we can examine the affect of the manifold on synthetic oversampling in a controlled environment on a large number UCI domains. The process of artificially augmenting UCI domains with a manifold property is discussed in the following subsection. In addition, the subsequent sections specify the algorithms, UCI datasets and evaluation methodology utilized in these experiments.

3.6.1 Manifold-Augmented UCI domains

The production of artificial manifold data is contingent on the common definition of manifold data that states that the probability mass resides in a lower-dimensional space. A simple way to synthesize this is to add columns to the data matrix that are not beneficial from a classification perspective (*i.e.*, force the probability density to reside in a lower dimensional space). We can achieve this by adding random variables, such as uniformly distributed random variables, that span both classes. In this case, we have selected a uniform distribution for its generality. In fact, any distribution would do just fine given that it is applied across both classes and is intended to carry no information. A synthetic oversampling method that is appropriate for manifold data will inherently detect the manifold and synthesize on it whilst less effective methods will synthesize in directions unassociated with the probability mass.

For some in the machine learning learning community, the way in which we artificially augment the UCI data to increase conformance to the manifold may be suggestive of a feature selection problem. Thus, we pause for a moment to address the similarities and important distinctions. In general, feature selection is not an effective means of solving problems involving manifold data. Feature selection methods, such as subset selection, simply choose a subset of the d dimensions to represent the data [Alpaydin, 2014]. A manifold space is a more general subspace that is formed from combinations of the original features. These combinations may be simple linear combinations:

$$f_i' = a_1 f_1 + a_2 f_2 + ... + a_d f_d, \tag{3.3}$$

where f_i' $i \in \{1, .., k\}$ is one of k components of the manifold-space embedded in the d-dimensional feature space; other manifolds are formed of much more complex non-linear combinations. In these cases, no subset of the original feature-space will represent the manifold. When the dataset conforms to the manifold assumption, a manifold learning method is required to extract a new set of k features from the d features in the original problem.

The manifold augmentation performed in these experiments is a simple linear form where $a_j = 0$ for $j \neq i$ and $a_j = 1$ otherwise. Feature selection could be beneficial on data with this synthetic manifold augmentation. The purpose of this simple formulation, however, is to demonstrate the degradation of the state-of-the-art in synthetic oversampling, and not to conduct a comparison with feature selection methods.

The process for adding the manifold is presented in Algorithm 1. It takes an n by m data matrix $\mathcal{X}$ and a percentage p as input and returns a new matrix $\mathcal{Y}$ which adds $m \times p$ columns with n rows to the original matrix $\mathcal{X}$ such that $\mathcal{Y} = [\mathcal{X}|\mathcal{Z}]$, where $\mathcal{Z}$ is the set of artificial manifold columns. Thus, $\mathcal{Z}$ is an n by $(p \times m)$ matrix of *i.i.d.* uniform random variables.

Algorithm 1 `produce-artificial-manifold`($\mathcal{X}$, p)

Input:

i) $\mathcal{X}$, an n by m dimensional binary data matrix.

ii) p, percent of a manifold as portion of dimensionality m.

Output:

i) $\mathcal{Y}$, the data matrix X with manifold columns added to it.

Method:

1: create $manifoldColumns = floor(p \times m)$ manifold conforming columns.
2: create an n by $manifoldColumns$ manifold matrix $\mathcal{Z}$.
3: let each column of $\mathcal{Z}$ contain m *i.i.d.* uniform random variables.
4: set $\mathcal{Y}$ equal to the column combined matrix $[\mathcal{X}|\mathcal{Z}]$
5: $Return(\mathcal{Y})$

End Algorithm

Using the above methodology, we compare the performance of the classifiers on the original UCI datasets ($p = 0$ in the Algorithm 1) to derivatives of the UCI datasets that have added manifold conformance based on $p = \{15\%, 30\%, 45\%, 60\%, 75\%, 90\%\}$. Thus, the derivatives contain the same implicit complexity as their parent datasets, with the added effect of the manifold. Therefore, any change in performance between $p = 0$ and $p > 0$ can be attributed to the increased presence of the manifold.

We demonstrate the increased conformance to the manifold assumption after manifold augmentation by plotting the eigenvalues of the target datasets before and after augmentation for the helix data distribution and the UCI breast cancer dataset in Figure 3.5. The upper plot depicts the eigenvalues produced on a three dimensional helix distribution, which is known to conform to the manifold assumption. The helix dis-

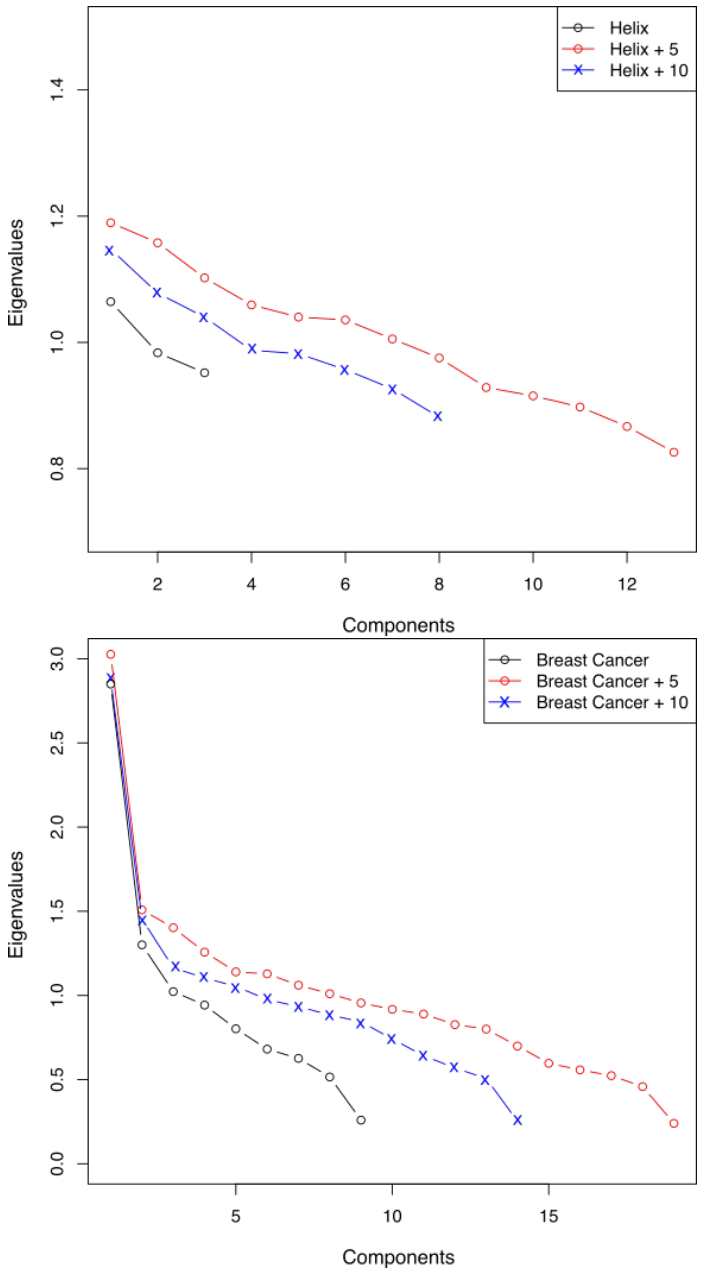


Figure 3.5: Manifold conformance to the helix distribution with with 0, 5 and 10 dimensions add (top) and manifold conformance to the breast cancer dataset with with 0, 5 and 10 dimensions add (bottom).

tribution is well-known to contain an embedded one-dimensional manifold. Using the helix allows us to see the effect of performing manifold augmentation via Algorithm 1 on an actual manifold distribution. For a more realistic dataset, the lower plot depicts the eigenvalues from the minority class of the UCI breast cancer dataset. In each case, the eigenvalues are plotted against the y-axis and the components are specified on the x-axis. The three components of the original helix data and the eight components of the original breast cancer data are plotted in black. The blue line plots specify manifold augmentation with an added five dimensions and the red line plots specify augmentation with an additional ten dimensions.

In the plot for the helix data, we see the manifold augmentation process continues the downward trend of the original components from the first component with a similar slope. This indicates that the manifold augmentation embedded the manifold deeper in the feature space. Similarly, we see the eigenvalues for the augmented breast cancer data follow the same trend as the original data. In each case, the added components have low eigenvalues, and thus, are not included in the implicit dimensionality. Therefore, we can say that we are increasing conformance to the manifold property.

3.6.2 Algorithms

Synthetic oversampling and the Weka implemented binary classification algorithms are applied in these experiments. Five prominent classification algorithms from the machine learning literature are applied in our study, namely support vector machine, multi-layer perceptron, C4.5 decision tree (J48)[3], Naïve Bayes and k-nearest neighbours. Each classifier was executed in the Weka machine learning environment with the Weka default parameters. Whilst it is very likely that we could obtain slightly better performance from each method with additional parameter tuning, this is not our objective. Rather, we are interested in the impact of the manifold on the data synthesization process. Thus, the default classification parameters are sufficient.

As previously discussed, our experiments focus on SMOTE, SMOTE+Tomek links and bSMOTE. Each of these methods has a single paramater, k, which dictates how many nearest neighbours to use. We ran our experiments for k values $3, 5, 7$. These are typical parameter values from the literature. A larger k value causes instances to be synthesized between more distant instances in the minority training set. The result is a more diverse set of synthetic instances, which can be very beneficial as the classifier is encouraged to

[3]The Weka implementation of C4.5 is called J48.

attribute a large region to the minority class, but also risks overgeneralizing the manifold.

3.6.3 Data

We utilize 16 UCI datasets in our experiments, each of which is selected to ensure a diverse range of dimensionalities over the datasets. Dimensionality is our primary criteria because of its relationship to the manifold assumption, and we select an assortment of datasets that conform to the manifold assumption with varying degrees. Table 3.1 specifies the selected datasets, along with the number of instances in the dataset, the dimensionality, the selected majority and minority classes and the total number of classes in the dataset. As we are interested in binary classification, we convert multi-class tasks by selecting a single class to form the minority class, and merging the remaining classes into one. In particular, we select the smallest class to form the minority class and merge the remaining classes.

For each experiment, we train on 25 minority training instances and 250 majority training instances; thus, we render each domain as a highly imbalanced classification task with the minority class less then 10% of the training set. More important than the class ratio is the concept of absolute imbalance (commonly seen as rarity in the minority class) put forth by [He and Garcia, 2009]. Whilst many classification problems exhibit extreme relative imbalance, if the training set is very large, it may not be a case of absolute imbalance. In scenarios marked by a class distribution of $1 : 500$, for example, we would say that this is a large imbalance. If we are given $100,000$ training instances, however, then could reasonably claim that this is not a case of absolute imbalance, and thus classifiers should be able to learn a good discriminant function [He and Garcia, 2009, Weiss, 2004, Jo and Japkowicz, 2004]. With this in mind, our focus is on understanding the performance under conditions of absolute imbalance, or rarity, where the literature shows class imbalance to be the most significant problem.

By setting each domain's training set to have the same class distribution, we control for the effect of sample size in the different domains and focus on the effect of the manifold. Because we want to maximize the number of instances that each algorithm is tested on in order to reduce variance in the results, we use the $a - priori$ class probabilities for the testing sets. Finally, each of these datasets is modified via Algorithm 1.

Table 3.1: This table presents the UCI dataset utilized in this chapter.

Dataset	Inst	Dim	Majority	Minority	Total Num Cls
BreastW	683	9	Benign	Malignant	2
Diabetes	768	8	Negative	Positive	2
Ecoli	336	7	0	1	2
Heart-statlog	683	9	Absent	Present	2
Ionosphere	351	34	g	b	2
Letter	20,000	16	¬R	R	26
Musk2	6,596	166	0	1	2
Opt Digits	5,618	64	¬4	4	10
ozone One hr	2,535	72	0	1	2
Pen digits	5,620	64	¬3	3	10
Segment	2,310	19	¬Brick-face	Brick-face	7
Sonar	208	60	Rock	Mine	2
Vehicle	846	18	¬Saab	Saab	4
WaveForm-5000	5,000	40	¬1	1	3
Yeast	1,483	8	¬MIT	MIT	10
Satlog	6,435	36	¬4	4	7

3.6.4 Evaluation

We use a specific form of re-sampling method for evaluation in this chapter due to the fact that we are modifying the class distribution in order to test our hypothesis regarding synthetic oversampling for imbalanced classification. We run a re-sampling procedure that is analogous to boosting 30 times. Specifically, at each iteration, we sample without replacement m minority class instances and n majority class instances, where m is significantly smaller than n. All remaining instances in the data set are placed in the testing set. This enables us to take advantage of all of the remaining data in our test sets at each iteration. Whilst we recognize that the overlap in training sets across iteration leads to a loss of absolute independence between iterations, we gain a significant benefit from the larger tests in terms of reduced variance in the results.

The common alternative to our approach is k-fold cross-validation, where the data is partitioned into k folds $\{f_1, f_2, ..., f_k\}$ of equal size. These methods are very useful for selecting the best classification method on target dataset but are not ideal for our purpose because they produce relatively small test sets. Moreover, if we were to stay true to their philosophy of independence between the tests in the cross validated runs, we would have to discard a large number of minority instances from the $k-1$ folds that form the training set at each iteration. Thus, for experimental work of this nature, we find the boosting-type method to be more suitable.

For each run of the boosting process, we record the AUC produced by each classifier. Our results are reported as the mean AUC produced over the 30 iterations of the boosting process.

3.7 Experimental Results

In this section, we report our findings on the impact of the latent manifold on the classification results produced after synthetic oversampling is performed. We demonstrate that existing synthetic oversampling methods are negatively impacted by data conforming to the manifold assumption. In doing so, we first develop a baseline for the impact of data with the manifold property on the five baseline classifiers. Subsequently, we examine the relative performance of these classification algorithms on the manifold data when synthetic oversampling is applied prior to classifier induction.

The sixteen UCI datasets specified in Table 3.1 are used in these experiments. In order to ensure absolute imbalance in the experiments we specify the minority training

set size to be $|TRN_{min}| = 25$. The artificial manifold adjustments, as described in Algorithm 1, are set to $p = \{0\%, 15\%, 30\%, 45\%, 60\%, 75\%, 90\%\}$, where $p = 0\%$ is the unchanged UCI data and $p = 90\%$ returns a modified dataset with the dimensionality increased by 90%.

The baseline analysis is presented in Section 3.7.1. This analysis servers to demonstrate the affect of the manifold on standard binary classifiers before synthetic oversampling has been applied. These results form our baseline. In Section 3.7.2, we compare the impact of the manifold property on the five binary classifiers after synthetic oversampling relative to the baseline. These sections will report a summary of the results whilst the complete table of results for these experiments are presented in Appendix A.1 and the corresponding plots are in Appendix B.1.

3.7.1 Baseline classification Analysis

In this section, we establish a baseline for classifier degradation (loss) to which we later compare the synthetic oversampling methods. The baseline results produced by the classifiers without synthetic oversampling are presented in Table 3.2. The first column specifies the dataset and the subsequent seven columns display the mean AUC produced by the five classifiers for p values between 0 and 90. The Final column is of greatest interest. It displays the degradation in performance from $p = 0$ to $p = 90$. Thus, the loss value for the letter dataset is $0.762 - 0.654 = 0.108$. This provides insight into how much of an affect the manifold property has on the baseline classifiers for each dataset.

This table illustrates that augmenting the UCI dataset with the manifold leads to the degradation of the baseline classifiers. Specifically, the AUC values for each data set decrease for increasing p-values. Letter, musk2 and segment suffer the largest loss of $0.108, 0.095$ and 0.087, whilst pima and breast have negligible losses of 0.006 and 0.003 respectively. Indeed, all of the dataset except diabetes and breast incur noteworthy losses as a result of the manifold. The diabetes and breast datasets are both very low dimensional and on the extreme ends of classification complexity. Diabetes represents a very complex classification task, and as a result a mean AUC of 0.568 was produced, whereas breast produces a rather simple classification task, thus resulting in an AUC of 0.915. We infer that the minimal impact results from the fact that a very challenging classification task results in a AUC that is close to 0.5, and therefore, cannot degrade very much. Alternatively, very easy problems that are linearly separable, for example, remain relatively easy. Moreover, as we show in Table 3.4, they both have weak initial

Table 3.2: This table presents the mean AUC results taken over the five baseline classifiers (without synthetic oversampling) on the artificially manifold-adjusted UCI datasets.

Dataset	0%	15%	30%	45%	60%	75%	90%	Loss
Letter	0.762	0.732	0.691	0.679	0.663	0.664	0.654	0.108
Musk2	0.724	0.690	0.668	0.650	0.642	0.630	0.629	0.095
Segment	0.864	0.825	0.802	0.794	0.787	0.787	0.777	0.087
Sonar	0.724	0.701	0.684	0.675	0.662	0.665	0.647	0.077
Pendigits	0.946	0.942	0.928	0.916	0.897	0.884	0.869	0.077
Ionosphere	0.778	0.766	0.751	0.746	0.738	0.723	0.718	0.060
Satlog	0.675	0.648	0.640	0.635	0.631	0.626	0.622	0.053
Ecoli	0.887	0.889	0.865	0.867	0.859	0.859	0.842	0.045
WaveForm-5000	0.657	0.649	0.641	0.631	0.626	0.624	0.619	0.034
Yeast	0.602	0.606	0.599	0.587	0.578	0.575	0.573	0.029
OZoneOnehrCols	0.625	0.615	0.613	0.603	0.605	0.599	0.598	0.027
Vehicle	0.581	0.569	0.566	0.559	0.564	0.557	0.559	0.022
OptDigits	0.828	0.826	0.819	0.824	0.819	0.820	0.812	0.016
Heart-statlog	0.755	0.750	0.756	0.751	0.744	0.745	0.742	0.013
Breast	0.915	0.920	0.920	0.906	0.907	0.912	0.909	0.006
Diabetes	0.568	0.565	0.562	0.562	0.559	0.560	0.565	0.003

manifold conformances. Each of these factors contribute to the fact that they do not show any significant degradation. Nonetheless, the fact that the performance degrades as a result of manifold augmentation is highly indicative of the need to apply machine learning methods appropriate for manifold data.

In the subsequent section, we compare these baseline loss values to the degradation incurred by the classifiers on the same datasets after synthetic oversampling is performed. This process establishes the explicit need for manifold-based oversampling methods.

3.7.2 The Impact of Synthetically Oversampling Manifold Data

The objective of this section is to consider the impact of the manifold property on synthetic oversampling. Our primary means for achieving this is to compare the loss values produced by the baseline classifiers in the previous section to the losses incurred as a result of the manifold property after synthetic oversampling.

The results presented in this section are a continuation of the experiments conducted in the previous section. Thus, the experimental setup is exactly the same, with the exception that synthetic oversampling is applied prior to classifier induction. Due to the very poor results produced by bSMOTE, we have only included k=3 for it. The complete table of results is included in Appendix A.1.

The loss results are presented in Figure 3.6 for SMOTE and SMOTE+Tomek with $k = 5$ along side the baseline loss. The other values of k can be viewed in Appendix B.1. This bar graph displays the difference between the AUC at $p = 0$ and $p = 90$. Therefore, a large loss value indicates a greater degradation. The bars in this plot serve to emphasize the many cases where a significant increase in loss value results from synthetic oversampling the manifold-augmented data. The musk2, satlog and vehicle datasets are good examples of this.

A more detailed view of the results is presented in Table 3.3. This table contains the baseline loss values and the loss values of the synthetic oversamplers with k values of $3, 5$, and 7. The performance on 11 of the 16 datasets degrades more compared to the baseline after synthetic oversampling. Examining the 96 instantiations of the synthetic oversamplers across the 16 datasets, we find that on 66 of the 96 occasions the oversamplers degrade more than the baseline. To accentuate the many cases where the loss is greater, we have shaded those cells grey. If we remove the 6 datasets (breast, ecoli, heart-statlog, ionosphere, pen digits and segmentation) on which the manifold had less impact on the synthetic oversamplers, we are left with 60 instantiations of the synthetic

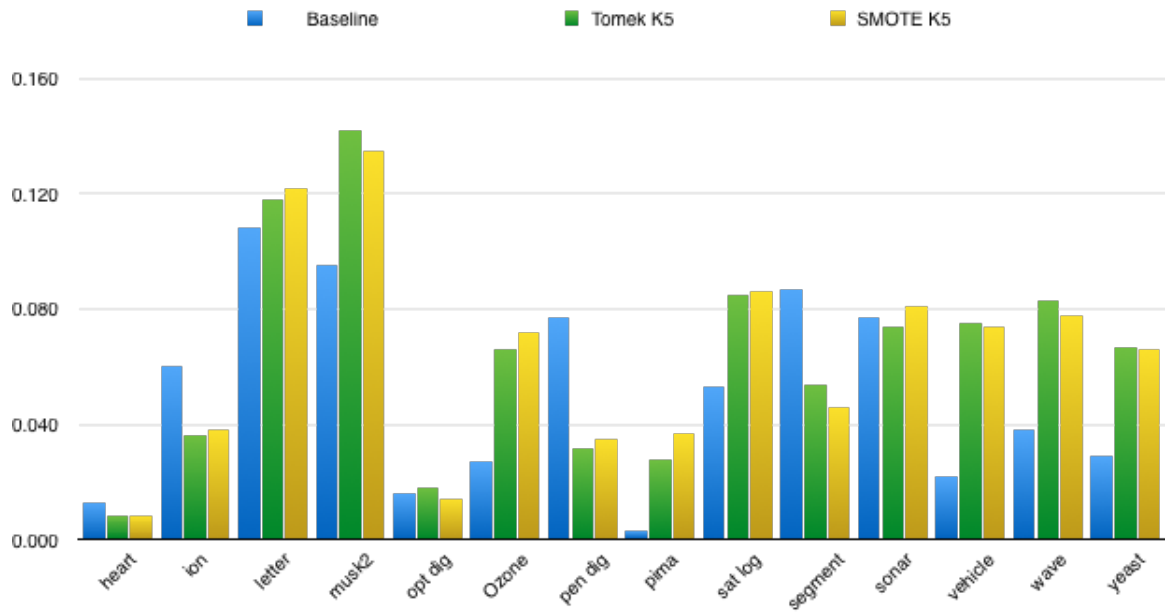


Figure 3.6: Comparison of the performance degradation of the mean of the baseline classifiers to SMOTE and SMOTE+Tomek links with K=5 when a manifold is artificially added to the UCI domains.

oversamplers applied to 10 datasets. Synthetic oversampling is heavily impacted by the manifold on these dataset regardless of the k value. On these datasets, 57 of the 60 instantiations of the synthetic oversamplers degrade more than the baseline. Moreover, the losses are large relative to the baseline. The loss on the 60 synthetic oversamplers for these datasets is on average 0.035 higher than the baseline. In practical terms this is, for example, as if suffering a degradation in AUC from 0.835 to 0.800, which is significant.

While it is not exclusive, it is clear that in the majority of cases the baseline classifiers degrade significantly more after the application of synthetic oversampling. These results are suggestive of our hypothesis that the state-of-the-art methods of synthetic oversampling do not synthesize manifold data very well, and as a result, the induced classifiers suffer. This provides good motivation for the consideration of manifold-based synthetic oversampling.

The few noteworthy cases where the synthetic oversampling does not degrade more than the baseline requires further investigation. As a means of understanding these special cases, we examine them in relation to their conformance to the manifold assumption. To do so we utilize exploratory factor analysis.

Exploratory factor analysis is a toolbox of methods for estimating the implicit di-

Table 3.3: This table presents the degradation between $p = 0$ and $p = 90$ of the baseline classifiers after the application of synthetic oversampling.

Dataset	Baseline	SMOTE K=3	SMOTE K=5	SMOTE K=7	Tomek K=3	Tomek K=5	Tomek K=7
Letter	0.108	0.129	0.122	0.126	0.122	0.118	0.116
Musk2	0.095	0.133	0.135	0.129	0.133	0.142	0.138
Opt Digits	0.016	0.020	0.014	0.014	0.029	0.018	0.018
Ozone 1hr	0.027	0.055	0.072	0.059	0.064	0.066	0.075
Pima	0.003	0.035	0.037	0.035	0.029	0.028	0.033
Sonar	0.077	0.082	0.081	0.082	0.077	0.074	0.088
Vehicle	0.022	0.070	0.074	0.079	0.062	0.075	0.081
Wave Form	0.034	0.074	0.078	0.078	0.080	0.083	0.080
Yeast	0.029	0.090	0.066	0.075	0.075	0.067	0.063
Satlog	0.053	0.093	0.086	0.087	0.088	0.085	0.088
Breast	0.006	0.003	-0.001	0.009	0.001	0.002	0.004
Ecoli	0.045	0.040	0.027	0.037	0.040	0.035	0.033
Heart-Statlog	0.013	0.010	0.008	0.008	0.015	0.008	0.008
Ionosphere	0.060	0.034	0.038	0.033	0.047	0.036	0.038
Pen Digits	0.077	0.035	0.035	0.033	0.038	0.032	0.031
Segment	0.087	0.051	0.046	0.040	0.050	0.054	0.054

mensionality of a dataset. These methods are particularly popular in the social sciences where the latent variables associated with the topic are either unknown are hard to measure directly. An important question from the field is how to best estimate the number of factors to retain [Courtney, 2013]. The traditional approaches to answering this question utilize the Kaiser criterion [Kaiser, 1960] and the scree test [Cattell, 1966b]. Given the practical importance of the field, methods continue to be developed. Whilst recent simulation studies have argued that more modern methods, such as parallel analysis [Horn, 1965] minimum average partial procedure [Garrido et al., 2011] and comparison data [Ruscio and Roche, 2012], are more accurate, the scree test and Kaiser criterion have remained prominent in modern research. For the purpose of manifold analysis in this thesis, we use the standard scree test. Our results suggest that, although it may not be sufficiently accurate for some studies in the social sciences, it produces reasonable estimates of conformance to the manifold property. Nonetheless, in we will consider a formal evaluation of the alternatives in our future work.

The scree test was proposed by Cattell as a means of determining the number of factors to retain in factor analysis or principle component analysis. He specifically noticed that when plotted in descending order of magnitude against their factors, eigenvalues level off at the point when the factors are primarily measuring random noise.

We use the scree test to estimate the implicit dimensionality of each dataset and divide it by the number of features:

$$M(D) = \frac{scree(D)}{dim(D)}, \tag{3.4}$$

where D is a given dataset. Therefore, a smaller $M(D)$ score suggests that the dataset conforms more strongly to the manifold assumption. We use the ratio in order to give preference to datasets that have a small scree value relative to the dimensionality. Specifically, a small scree value in relative high-dimensional tasks indicate conformance to the manifold assumption.

The first factor that we find to contribute to the fact that six of the datasets (breast, heart, ecoli, ionosphere, pen digits and segment) do not incur a greater loss after synthetic oversampling has to do with the overall complexity of the classification tasks. These are the easiest dataset to classify according to the baseline AUCs. Thus, we find that if the classification is particularly easy, then the manifold augmentation has less of an impact. In general, we can extend this finding to surmise that if the two classes are disjunct, then a classifier is capable of performing well in spite of the manifold.

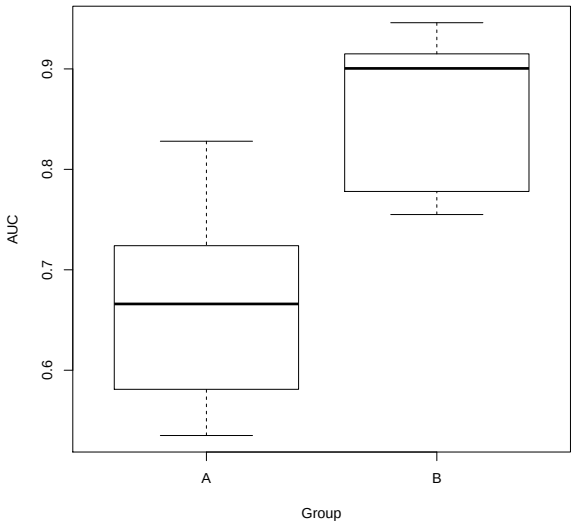


Figure 3.7: Box plot of the AUC performances for datasets with a greater degradation (group A) and less of a degradation (group B) due to the manifold. This suggests that the manifold has less impact when the classes are easily separable.

To emphasize the performance differences in our two cohorts of classification task (more and less affected by the manifold), Figure 3.7 displays the AUC performances for datasets with a greater degradation (group A) and less of a degradation (group B). These are shown in the form of a box plot. It is clear from this plot that the small group of datasets where synthetic oversampling does not cause a greater degradation after manifold augmentation (group B) are simply not challenging classification tasks, and as a result, the manifold has minimal impact.

The degree to which the original data $p = 0$ conforms to the manifold assumption based on the $M(D)$ score also has an impact on degradation, as well. Naturally, this is accentuated by the degree to which the augmented dataset D' conforms the manifold assumption. The $M(\cdot)$ scores for each UCI data used in the experiment are displayed in Table 3.4. The datasets are sorted in increasing order so that those that conform the most to the manifold assumption appear at the top of the list. Of the six datasets that do not have increased degradation with synthetic oversampling, only the segmentation dataset does not rank near bottom for conformance according to $M(\cdot)$. Thus, their manifold conformance is relatively low compared to the other datasets and the complexity is also

low, and, as a result, synthetic oversampling is not negatively impacted.

Table 3.4: Manifold conformance score based on $M(\cdot)$ for each test dataset.

Dataset	$M(D)$
Wave-form	0.025
Ozone One	0.083
Sonar	0.100
Segment	0.105
Musk	0.108
Vehicle	0.111
Satlog	0.139
Opt Digits	0.172
Letter	0.188
Ionospher	0.206
Breast	0.222
Pen Digits	0.250
Diabetes	0.375
Heart	0.385
Ecoli	0.571
Yeast	0.625

Interestingly, diabetes and yeast have low conformance to the manifold with respect to $M(\cdot)$; thus, they appear at the bottom of the list of conformance to the manifold assumption based on their $M(\cdot)$ scores. We might infer from this that the degradation will be low; however, this is not the case. What we find is that even mild conformance to the manifold property can have a significant impact on synthetic oversampling when the data has other complexities such as overlap and multi-modality. The baseline complexities of these two datasets are the highest, as suggested by the AUC values of 0.569 and 0.602, respectively, and we find that their corresponding degradations as a results of manifold augmentation are also high. Therefore, the impact of the manifold on synthetic oversampling is controlled by:

- Conformance to the manifold assumption; and
- The complexity of the classification task.

The positive aspect of this discovery is that the manifold property leads to increased degradation is that a net benefit of synthetic oversampling in terms of the AUC, nonetheless, exists. Therefore, this increased level of degradation after synthetic oversampling provides a considerable amount of potential in terms of performance gains. An appropriate manifold oversampling method has the potential to provide the same advantage on imbalanced data, and maintain it when the manifold property exists. We demonstrate a means of achieving this potential in the subsequent chapter.

3.8 Conclusion and Future Work

Manifold-based methods are often considered essential to achieving good results on many important machine learning domains. In spite of its prominence in the wider field of machine learning, the manifold property has not received any attention within the context of class imbalance and synthetic oversampling. This chapter serves to raise awareness of the need to study and understand the relationship between synthetic oversampling methods and the data that they are applied to. This fact is of particular importance given that these methods are required to make large generative leaps from small training sets on imbalanced classification tasks.

Our results show that although the existing methods of synthetic oversampling improve the AUC beyond that of the baseline classifiers, they are not reaching the full potential of synthetic oversampling on manifold data. Specifically, we show that in general the performances of the baseline classifiers degrade more when they are applied in conjunction with the existing synthetic oversampling methods, which have been designed without consideration given to the relationship between their individual biases and the data to which they are applied. This fact motivates our introduction of a manifold-based synthetic oversampling framework in the subsequent chapter.

Although we have demonstrated the impact of the latent manifold on the existing synthetic oversampling methods in general, manifolds are complex structures and can take a variety of forms. A major future work is to discover and attempt to categorize the different properties of manifold data in a machine learning context and to further understand the relationship between these properties and synthetic oversampling methods.

Chapter 4

MOS: A Framework For Synthetically Oversampling the Manifold

4.1 Introduction

In the previous chapter, we demonstrated the detriment of ignoring the existence of the manifold property when performing synthetic oversampling to address class imbalance. Pervious work in machine learning has similarly identified negative impacts of the manifold property on general purpose classification and clustering algorithms, and have successfully applied manifold-based methods for these tasks [Zhang and Chen, 2005, Weinberger and Saul, 2004, Crawford and Ghosh, 2005, Tuzel et al., 2007, Yu et al., 2009]; thus, it is with this foundation that we champion the idea of manifold-based synthetic oversampling in this chapter. Moreover, we recognize that the breadth of methods within the field of manifold learning utilize diverse assumptions and biases, and thus, to limit our solution to a single manifold learning algorithm would be to place a limitation on the grand potential of our idea. As a result, in this chapter we develop a framework for manifold-based synthetic oversampling. We demonstrate two formalizations of it, and show that manifold-based synthetic oversampling outperforms the more naïve, generic approaches that are currently considered to be state-of-the-art.

4.2 Motivations

The previous chapter demonstrated the weakness of the state-of-the-art methods in synthetic oversampling data that conforms to the manifold assumption. Specifically, we saw that the existing methods cause a degradation in the AUC performance when instances are synthesized from data with the manifold property. Given that data conforming to the manifold assumption is quite predominant in real-world settings, such as image and text classification, sensor data and spectral data, we consider it to be of great importance to develop a method that can synthesize instances in a manner that is appropriate for such data.

4.3 Contribution

This chapter contains the primary contribution of the thesis. In particular, we a) define a general framework for synthetically oversampling the latent manifold, *b)* demonstrate two formalizations of the framework using principle component analysis (PCA) and a denoising autoencoder (DAE), including how to generate samples from the induced manifold and *c)* show that the proposed method outperforms the state-of-the-art in synthetic oversampling on domains that conform to the manifold assumption.

4.4 Framework

4.4.1 Overview

If we know a specific property of our target data, then to ignore it and apply a general purpose algorithm is to put ourselves at a significant disadvantage. When modeling human height, which is approximately a Gaussian processes, applying a Gaussian form will produce the best results. And so it is that when we know that the manifold property exists in classification and unsupervised learning data, using a manifold learning method in the solution leads to superior results [Zhang and Chen, 2005, Weinberger and Saul, 2004, Crawford and Ghosh, 2005, Tuzel et al., 2007, Yu et al., 2009]. It is with this insight that we propose that the solution to imbalanced classification problems also benefits from manifold learning.

Manifold learning is an important topic of study in mathematics and computer science, and as a result, many methods have been proposed for inducing models of the

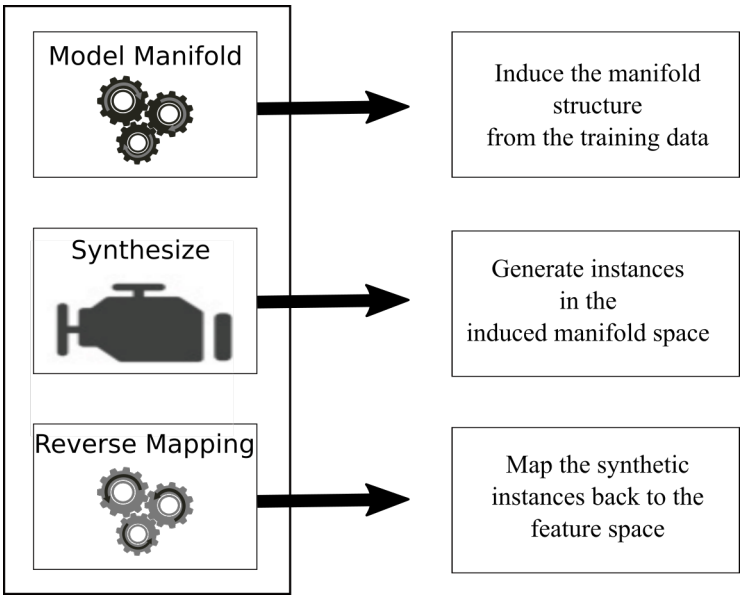


Figure 4.1: General framework for synthetically oversampling.

latent manifold [Huo et al., 2007, Ma and Fu, 2011]. Each of these methods makes a different set of assumptions about the underlying manifold space; PCA, for example, assumes linearity and normally distributed variables. Alternatives, such as the one based on DAE, allow for non-linearity. The biases inherent in each method allows us to select the one most appropriate for a given problem. For this reason, we propose a framework that facilitates the use of whichever manifold learning technique is most appropriate for the given task.

A schema of the framework for manifold-based synthetic oversampling is presented in Figure 4.1. The framework is composed of three parts. The first element of the framework induces a manifold representation of the training data. The modelling can be performed with any manifold learning method, such as PCA, kernel PCA, DAE, local linear embedding, *etc.* Data is synthesized along the induced manifold during the second phase of the framework, and the final phase maps the synthesized data to the original feature space where classifier induction is performed.

The second and third phases of the framework are dictated by the choice of manifold learning algorithm in the first phase. If our learning objective involves a linear manifold, or the training data is extremely rare, PCA is an effective means of modelling. PCA produces good results because, in spite of the fact that manifold learning reduces the number of examples needed to model a distribution, scenarios arise in class imbalance

where a simple model is the best option. Alternatively, non-linear problems with more training data are well suited for modelling via DAE. Our experiments focus on PCA and denoising autoencoder because together they can model linear and non-linear manifold that are simple or complex. Moreover, they offer effective and easy-to-implement means of sampling from the induced manifold. Nonetheless, local linear embeddings, ISOMAP and other manifold learning techniques can be used when they are preferable [Huo et al., 2007, Izenman, 2011].

4.4.2 Formalizations

Formalization with PCA

PCA is, perhaps, the most commonly used means of modelling a latent manifold structure. In addition, it has a simple mathematical form that is ideal for demonstrating the strengths of the framework, along with the specifics of each component of the framework. The simplicity of PCA, however, is not a limitation with respect to synthetic oversampling, but a strength in certain cases. In particular, PCA is an appropriate choice when:

- The minority class is embedded in a linear manifold; and/or
- The minority class is very rare.

The latter point can be seen as the application of Occam's razor, which argues that simple solutions generalize better, to synthetic oversampling. Inducing a complex model from few training points has a high likelihood of overfitting. With this in mind, a simple manifold is a better assumption than a complex non-linear manifold when training points are very rare.

PCA is a linear mapping from the d-dimensional input space to a k-dimensional transformation space where $k \ll d$. The process involves first calculating the d-dimensional mean μ and $d \times d$-dimensional covariance matrix $\mathbf{\Sigma}$. Subsequently, the eigenvectors E and eigenvalues λ are calculated and sorted in decreasing order according to the λ values. The $\mathbf{e}_{1...k}$ associated with the k largest λ are taken to represent the transformed space. The k value is typically selected to include the few large eigenvalues; beyond k, the eigenvalues tend to level off and are assumed to be associated with random noise.

For synthetic sampling we aim to utilize the induced model of the latent manifold to accurately generate instances in the physical space of the data. Thus, all of the k components of the PCA are informative in the generative process; however, they are not all equal. We use the λ values to control the spread along the principle components.

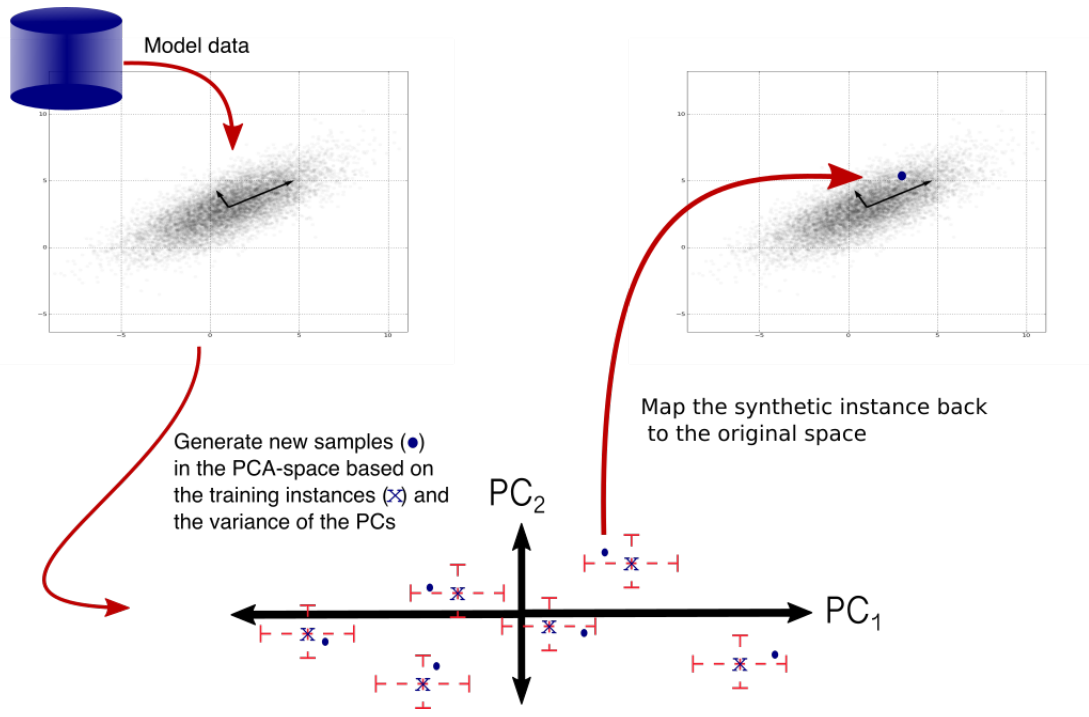


Figure 4.2: The process of synthesizing instances via PCA

For the PCA realization of the framework, we first induce a PCA model $pca = \{\mu, \Sigma, E, \lambda\}$ of the target class $T_{m\times d}$, where m is the number of instance and d is the dimensionality. According to the second aspect of the framework, sampling is performed in the PCA-space based on the induced model.

Whilst there is no prescribed method for sampling a PCA model in the literature, we can envision a few options. Here, we produce a synthetic set $S_{n\times d}$, where n represents the desired number of synthetic instances. The synthetic instances are produced by randomly sampling n instances from T' (T in the pca-space) with replacement and applying *i.i.d.* additive Gaussian noise $\mathcal{N}\big(0, \Sigma\big)$ to each instance in S. Σ is a diagonal matrix with each $\sigma_{i,i}$ in Σ selected based on λ_i in order to promote spread along the principle components associated with greatest variability and minimize spread along the others. Finally, the synthetic set S is mapped into the set $S' = S \times E^{-1}$. This process is depicted in Figure 4.2 for a two-dimensional Gaussian problem and detailed in Algorithm 2.

The classification tasks that we often face are non-linear. In these cases, we prefer a manifold learning technique that facilitates the induction of a possibly complex non-linear manifold. We utilize the denoising autoencoder for these cases because it can

Algorithm 2 `pca-SyntheticOversampling`(X, n, β=1)

Input:

i) $\mathcal{X}$, an m by d dimensional data matrix.

ii) n, the number of instances to synthesize.

iii) β, scaling factor applied to the spread along the principle components .

Output:

i) $\mathcal{Y}$, the synthetic samples.

Method:

1: $colMean = columnMeans(\mathcal{X})$: Calculate column means.

2: $\mathcal{X} = \mathcal{X} - colMeans$: Centre the data.

3: $\Sigma = cov$: Calculate the covariance matrix of $\mathcal{X}$.

4: $\lambda = eigenValues(\mathcal{X})$: Calculate the eigenvalues.

5: λ_{norm}: normalize the eigenvalues between $[0, 1]$.

6: $E = eigenVectors(\mathcal{X})$: Calculate the eigenvectors.

7: $\mathcal{Y}$: n instances of $\mathcal{X}$ sampled with replacement.

8: $\mathcal{Y} = E \times \mathcal{X}$: Map X to PCA-space.

9: $\mathcal{Y}' = \mathcal{Y} + \mathcal{N}\big(0, (\lambda_{norm})/\beta\big)$: Apply noise with zero mean and λ_{norm} standard deviation scaled by β to $\mathcal{Y}$.

10: $\mathcal{Y} = E^{-1} \times \mathcal{Y}'$: Map Y to the target data-space.

11: $\mathcal{Y} = \mathcal{Y} + colMeans$: Undo the centring of the data.

12: $Return(\mathcal{Y})$

End Algorithm

model a non-linear manifold, and we can directly control the complexity of the manifold by adjusting the size of the hidden layer. This is advantageous because when only a small number of minority training instances are available, we must prevent the model from becoming overly fit on the few examples. We do this by inhibiting the complexity of the model by specifying a small number of hidden units relative to the dimensionality of the domain.

Formalization with DAE

DAE Fundamentals: A young boy is given a variety of different candies. Having seen, smelled and tasted these candies, the child develops a general mental model of what it means to be a candy. The specific form of the model is, however, entirely hidden from the boy and all others. This is inconsequential in the sense that so long as the model exists, the boy will be capable of performing the essential task of recognizing these and other completely different candies in the future.

In much the same sense, a denoising autoencoder learns a hidden representation of its target class. Stimulus enters the network through an input layer as it may enter the human brain through the eyes or ears, and is stored in the form of a distinct hidden representation that can be recalled to its initial form with minimum loss of information.

Through this learning process, the network creates a mapping from the input-space to the hidden-space which incorporates the bias in which the hidden layer represents the manifold-space of the target data. The second aspect of learning a denoising autoencoder involves optimizing the set of weights that map the stimulus from the hidden representation back to its original form with minimal loss of information. A graphical example of a denoising autoencoder network is depicted in Figure 4.3.

More formally, regularized autoencoders, such as contractive auotencoders and denoising autoencoders, have been studied in relation to their inductive abilities on domains involving lower-dimensional manifolds [Rifai et al., 2012, Alain and Bengio, 2014]. It has been demonstrated that these methods capture the local manifold structure through the leading singular vectors of the Jacobian [Rifai et al., 2012]. Alain and Bengio articulate the mechanics of denoising autoencoders as requiring that the reconstruction function be as simple as possible, and examples that are neighbours on the high-density manifold be represented differently [Alain and Bengio, 2014]. Thus, accurate reconstruction of points on the manifold is made possible, whilst the reconstruction error quickly rises for examples orthogonal to the manifold. This is the essential fact that initially led us to consider DAEs as a means of synthetic oversampling. Specifically, the inherent property

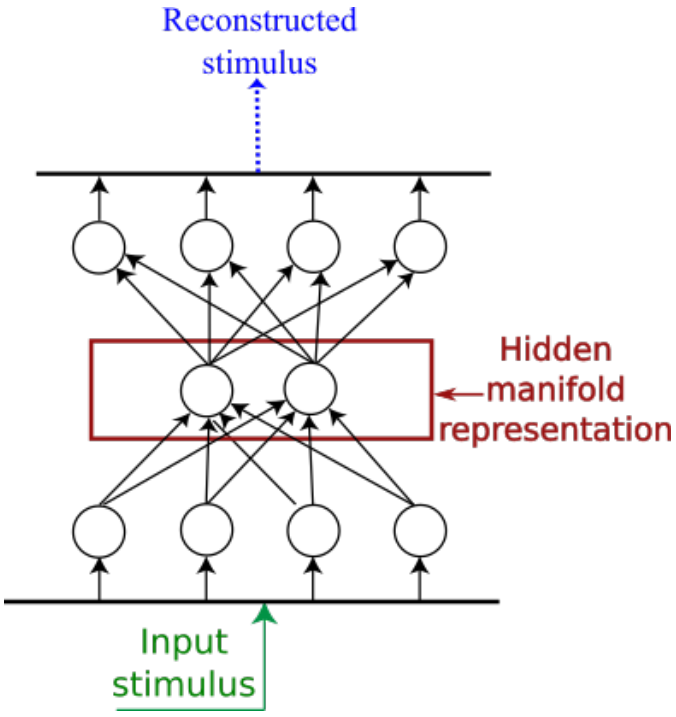


Figure 4.3: General form of an autoencoder.

of a denoising autoencoder is that, given a vector that has been sampled away from the manifold, they can map the vector orthogonally to the manifold. Therefore, we can take an arbitrary distribution in the target feature space and utilize a trained denoising autoencoder to map instances from this distribution to the induced manifold of the target class; the results of the mapping can then be taken as synthetic samples of the manifold.

The fundamental components of the autoencoder are its encoding $h = f_\theta(x)$ and decoding functions $g(\cdot)$. The encoding function $h = f_\theta(x)$ is composed of

$$f_\theta(x) = s(\mathbf{W}x + b) \tag{4.1}$$

where the θ represents the parameter set $\{\mathbf{W}, b\}$, $\mathbf{W}$ is a $d \times d'$ weight matrix and b is a d-dimensional bias vector. The function s, is a non-linear squashing function, such as the sigmoidal or tanh. Subsequent to encoding, the d'-dimensional encoding is decoded via $g(\cdot)$ where,

$$g_{\theta'}(y) = s'(\mathbf{W}'y + b'). \tag{4.2}$$

Here, $\mathbf{W}'$ and b' represent the weight matrix and the bias vector that cast the encoded vector back to the original space. The s' function is typically linear in autoencoders. The act of encoding/decoding alone, however, does not guarantee that the network will learn

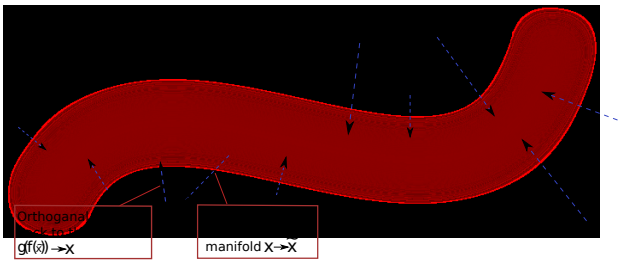


Figure 4.4: Demonstration of the mapping of examples to the manifold (in red). In this example of handwritten fours, noisy, or non-fours, that are off the manifold are mapped orthogonally to the induced manifold via the function $g(f(\cdot))$. The result is an example of a new four on the manifold.

something interesting. In the most trivial of cases, for example, one could envision the network producing a copy of the input at the output layer; such an extreme overfitting is unlikely to generalize well [Vincent et al., 2010]. To avoid this, the early work utilized compression networks to force the network to have better balance of generality and specificity. More recently, sparse encodings have been popularized as a method to facilitate the use of expansion networks [Olshausen and Fieldt, 1997, Vincent et al., 2008, Srivastava et al., 2014]. These networks apply sparsity to the hidden layer by turning on/off nodes in the hidden layer or input layer, or by applying noise to the input vector. In doing so, each method prevents overfitting. Through the study of deep learning, denoising autoencoders have appeared as an effective approach to force an expansion network to learn interesting features.

Denoising autoencoders are trained by passing corrupted versions of the target data $\tilde{x}$ into the network with the aim being that the denoising autocoder learns to reconstruct the uncorrupted version of the input x. Specifically, they are trained to minimize the squared error between x and output $g(f(\tilde{x}))$ over the training set via gradient descent with backpropagation. In our case, $x \in \mathbb{R}^n$, therefore we apply Gaussian corruption $\tilde{x}|x \sim \mathcal{N}(x, \sigma^2\mathcal{I})$ with zero mean and unit standard deviation. This corruption function is applied to the original x at each epoch of the training process. In this manner, the learning process forces the denoising autoencoder to extract the essential characteristics of the training data.

Synthetic Oversampling DAE: Denoising autoencoders are applied to induce a nonlinear manifold of the minority class from which synthetic samples are drawn to populate

Algorithm 3 `dae-model-selection`(X, $epRange$, $huRange$, $sigRange$, $numRandModels$)

Input:

i) $\mathcal{X}$, an m by d dimensional data matrix.

ii) $huRange$, an upper and lower bound on the number of hidden units.

iii) $epRange$, an upper and lower bound on the number of epochs.

iv) $sigRange$, an upper and lower bound on the variance of Gaussian noise.

v) $numRandModels$, the number of random models to test.

Output:

i) $DEA_{\{\mathbf{W},b\}}$, a trained denoising autoencoder with weight matrix $\mathbf{W}$ and bias b.

Method:

$\mathcal{X}'$: column normalization of X between $[-1, 1]$
$curBestError = \infty$
$curBestDAE = NULL$
for 1..$numRandModels$ **do**
  hu: uniform random int in range $huRange$
  ep: uniform random int in range $epRange$
  σ: uniform random real in range $sigRange$
  $\mathbf{W}$: initial random weight matrix
  b: initial random bias vector
  $DAE_{\{hu,ep,\sigma,\mathbf{W},b\}}$: initialize DAE
  $DAE_{\{\mathbf{W},b\}} = train(DAE_{\{hu,ep,\sigma,\mathbf{W},b\}}, \mathcal{X}')$ with early stopping and random resets
  $evalRes = \frac{1}{|\mathcal{X}|}\Sigma\left[\mathcal{X}' - DAE_{\{\mathbf{W},b\}}(\mathcal{X}')\right]^2$
  if $curBestError \geq evalRes$ do
    set $curBestError$ to $evalRes$
    set $curBestDAE$ to DAE
  end if
end for
$Return(DEA_{\{\mathbf{W},b\}})$

End Algorithm

the minority training set. Figure 4.4 provides a generic representation of how a trained autoencoder maps input vectors (sample initiation points) to the induced manifold (at the end of the arrows). A concrete demonstration is provided with the handwritten 4s. The sample initiation points are produced by corrupting a training instance. The result is a blurry 4 that is distinct from its source and off the manifold. This can be seen at the start of the arrow in the figure. The sample initiation 4 is then mapped orthogonally back to the manifold representation that was induced during training via $f(g(\tilde{x}))$. Therefore, the corruption process $x \rightarrow \tilde{x}$ is a non-orthogonal mapping of a training instance off the manifold, and $f(g(\tilde{x}))$ is an orthogonal mapping back to the manifold. In this way, we can produce infinitely many distinct samplings of the induced manifold. Metaphorically, one can think of the orthogonal mapping to the manifold to be akin to the sun's gravitation pull on the surrounding celestial objects. The trained weights of the denoising autoencoder cause the unrepresentative points to be pulled onto the manifold, thereby rendering them representative of the latent manifold.

Mathematically, the sample initiation points take the form of input vectors $\tilde{x}$ and the synthetic samples x are the result of passing $\tilde{x}$ through the trained DEA function $x = g_{\theta'}(f_{\theta}(\tilde{x}))$. In Figure 4.4, the denoising autoencoder learns the latent manifold (represented in red) and is then used to map a set of sample initiation points that are spread around the manifold, orthogonally to the manifold. In this way, we produced synthetic instances, such as the sampled handwritten 4 shown in the figure.

The complexity of the induced manifold can be controlled based on the parameters of the DEA. More hidden units and longer training times, for example, will lead to a more complex manifold. From the perspective of synthetic oversampling, a simple manifold is typically desirable due to the limited number of training examples.

For simplicity, we have divided the algorithm into two separate processes. Algorithm 3 takes a target training set, and user-specified upper and lower bounds for the hidden layer, training epochs, and variance on the Gaussian noise. From these parameters a user-specified number of random models are constructed, trained and evaluated based on the mean reconstruction error and the best model is returned. The selected model, training data and the desired number of synthetic samples are then passed into Algorithm 4 as parameters.

In Algorithm 4, the sample initiation points $\mathcal{Z}$ are constructed by applying Gaussian noise to the normalized target data $\mathcal{X}'$. The sample initiation points $\mathcal{Z}$ are then mapped orthogonally to the manifold via the induced denoising autoencoder function $g(f(\mathcal{Z}))$. The output, $\mathcal{Y}$, of the algorithm is denormalized and returned for use in inflating the

Algorithm 4 `dae-SyntheticOversampling`($\mathcal{X}$, $DAE_{\{\mathbf{W},b\}}$, n, σ)

Input:

i) $\mathcal{X}$, an m by d dimensional data matrix.

ii) $DEA_{\{\mathbf{W},b\}}$, a trained denoising autoencoder with weight matrix $\mathbf{W}$ and bias b.

iii) n, the number of instances to synthesize.

iv) σ, variance of the Gaussian sample initiation noise.

Output:

i) $\mathcal{Y}$, the synthetic samples.

Method:

1: $\mathcal{X}'$: column normalization of $\mathcal{X}$ between $[-1, 1]$.

2: $normParams$: column normalization parameters of $\mathcal{X}$.

3: $\mathcal{Z}$: normalized $\mathcal{X}$ plus sample initiation noise $\mathcal{N}(0, \sigma)$.

4: $\mathcal{Y}' = DAE_{\{\mathbf{W},b\}}(Z)$: samples $\mathcal{Y}'$ from the induced manifold.

5: $\mathcal{Y}$: denormalization of $\mathcal{Y}$ based on $normParams$.

6: $Return(\mathcal{Y})$

End Algorithm

minority training set.

Alternatives

We have thus far demonstrated two preferential formalizations of our proposed framework for manifold-based synthetic oversampling. As we have previously stated, however, there is a great breadth of research in the general field of manifold learning [Huo et al., 2007, Ma and Fu, 2011, Izenman, 2012]. For a given imbalanced classification task, the practitioner may find that the biases and assumptions inherent in a certain manifold learning method are preferable to the methods that we have demonstrated. For this reason, we emphasize that the framework is intended to be flexible enough to encourage the consideration and selection of the most appropriate manifold learning method.

4.5 Demonstration

An ideal way to demonstrate the competitive advantage of manifold-based solutions to synthetic oversampling is to visualize instances synthesized for some common manifolds. Handwritten digits provide a very practical manifold learning domain that is easy to

visualize. The main challenge with handwritten digits is in the subjectiveness of visually evaluating the results. Thus, we include the helix and the swiss roll domains to further emphasize the distinction. The helix and swiss roll are more theoretical domains that provide a very clear perspective on the performance of the synthetic oversampling methods [Xue and Chen, 2007, Jia Wei et al., 2008, Goldberg et al., 2009, Silva and Tenenbaum, 2002, Weinberger et al., 2004].

4.5.1 Handwritten Fours

Handwritten 4s from the *minst* dataset provide a practical manifold learning task. Each training 4 is drawn from a 28×28 grey-scale image. Image learning problems, such as facial recognition and character recognition, conform to the manifold assumption in the sense that the target object exists in a subspace of the $M \times N$ pixel image. To understand this, consider that there are infinitely many random combinations of grey-scale pixels in the feature space that do not make 4s. The task of the manifold learner is to infer the subspace where the fours exist. In this space, we are much more likely to synthesize or classify correctly.

A random selection of 16 fours that were generated by the manifold-based synthetic oversampling framework with the denoising autoencoder formalization and the PCA formalization are presented in Figure 4.5. In addition, we report results for SMOTE and the kernel-based method. Each of these systems was trained on the same 25 handwritten fours. The objective is to synthesize instances that look like well constructed 4s and to synthesize distinct synthetic 4s. Producing replicas of a single, very nice, four is not sufficient. The kernel-based approach is very clearly a poor generator of fours. Though the shape of the fours can be seen, they are very blurry. This blurriness of the fours indicates a spreading away from the manifold resulting from the fact the synthesization is performed in the feature space.

Twelve of the sixteen fours produced by SMOTE are well constructed. Many of these, however, are skewed to the left. We refer to the style of the 4s in the exceptionally bad cases as *stacked 4s*. We demonstrate this in the figure as generating a new four by placing two very different fours on top of each other. Once again, this results from the fact that SMOTE performs synthesization in the feature space, rather than the manifold space, and generates new fours between two training fours. If the space between the 4s is on the manifold, a good four is produced. When the data conforms to the manifold assumption, however, the distanced-based approach utilized by SMOTE is likely to fail due to the

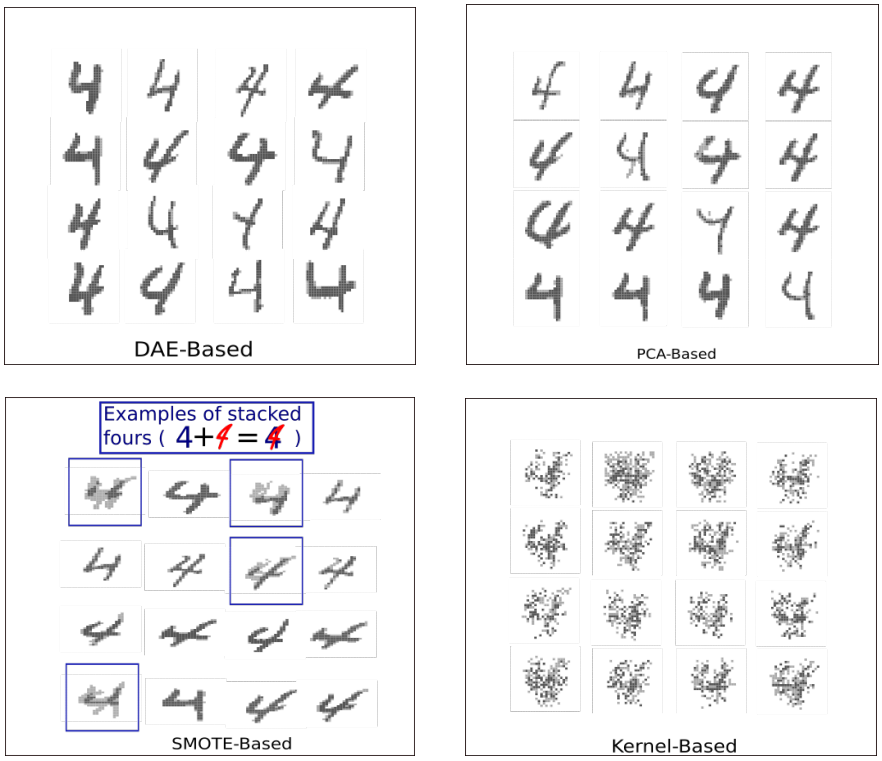


Figure 4.5: From left to right, handwritten fours synthesized by DAE, PCA, SMOTE and kernel-based methods.

fact that the straight-line distance cannot account for the curvature of the manifold, and is, at best, only appropriate within each instance's local neighbourhood. Given a small minority training set, however, it is unlikely that the kNN-set of any instance will reside in the same local neighbourhood. Thus, there is an elevated risk of synthesizing instances away from the manifold.

PCA produces reasonable 4s. Only three of these are of low quality and none are skewed in anyway. All but one of the fours produced by the denoising autoencoder are well constructed; moreover, denoising autoencoders produce a very good amount of diversity in the set of 16 fours.

These results illustrate the relative advantage of manifold-based synthetic oversampling on a practical and high-dimensional domain, but they leave some subjectiveness in the evaluation. We can all agree on the well constructed 4s and those that are poor examples of 4s, but there is a foggy middle ground. The lower dimensional helix and swiss roll domains reduce the subjectiveness of our evaluation.

4.5.2 Helix

The helix distribution effectively illustrates the strength of our method, as the three-dimensional helix can be represented via a one-dimensional linear manifold. For this reason, the helix distribution is consistently utilized for demonstration purposes in manifold learning [Xue and Chen, 2007, Jia Wei et al., 2008, Goldberg et al., 2009, van der Maaten et al., 2009, Feuersänger and Griebel, 2009, Sparks and Madabhushi, 2013]. This experiment demonstrates that the existing methods of synthetic oversampling produce an excessive spread of synthetic instances away from the manifold. In order to increase the difficulty of the modelling task, and emphasize the strength of the proposed framework, a dataset of the form $H = h(\cdot) + \mathcal{N}(\cdot)$, where $h(\cdot)$ samples a pure helix and $\mathcal{N}(0, 0.1)$ samples a Gaussian distribution with zero mean and 0.1 standard deviation, is deployed. The pure helix is defined as $x_1 = rcos(t)$; $x_2 = rsin(t)$; $x_3 = ct$, where $t \in [0, 2pi)$, r is the radius of the helix and $2\pi c$ is a constant specifying the vertical separation of the loops.

Figure 4.6 plots the training data and the synthetic data produced by the denoising autoencoder and PCA formalizations for the framework along with SMOTE and a kernel-based synthetic oversampling solution. The circles in the figure represent the training data and the triangles are the instances synthesized by each method. The red boxes highlight the weaknesses of each method.

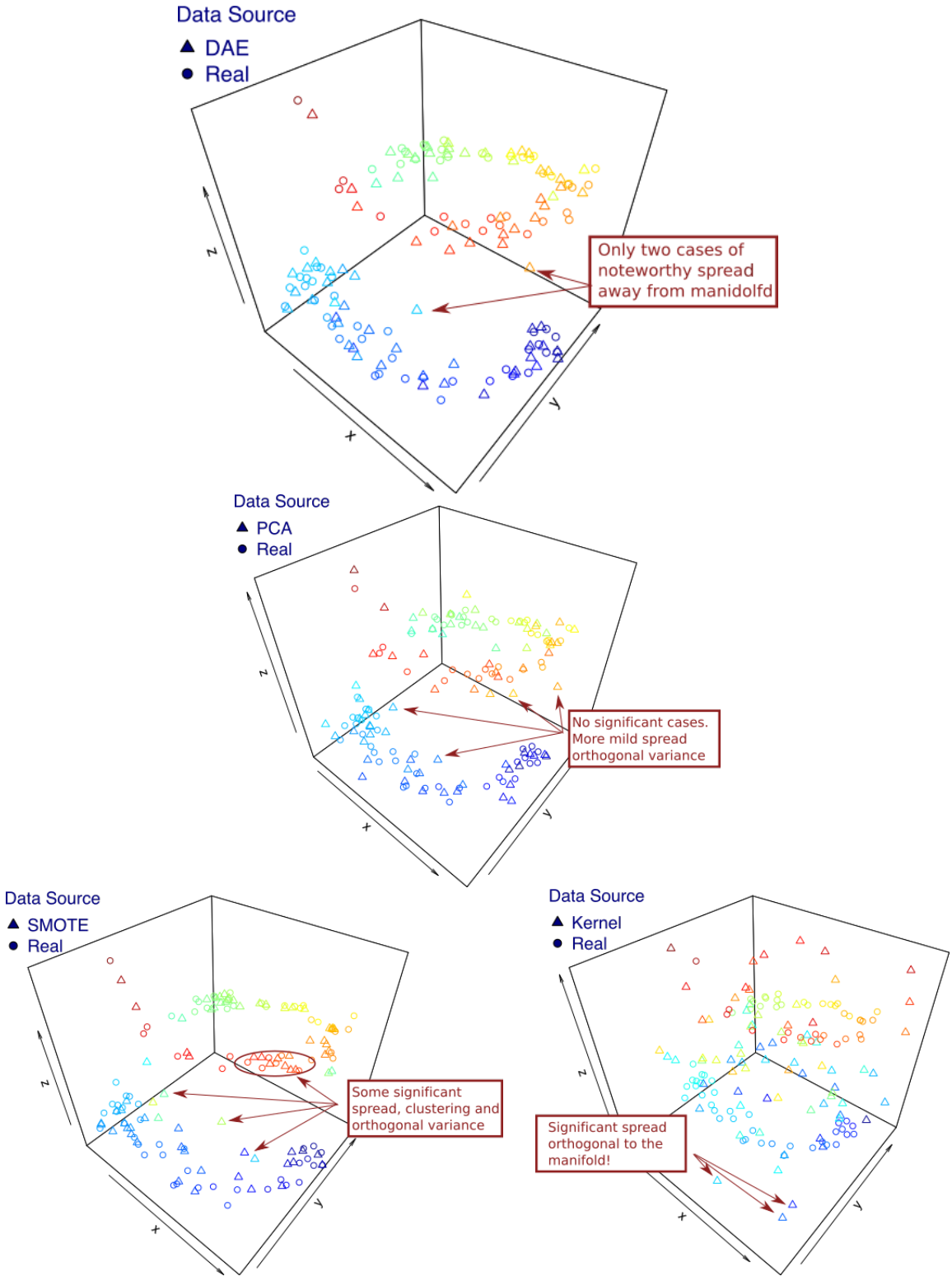


Figure 4.6: From left to right, helix data synthesized by DAE, PCA, SMOTE and kernel-based methods.

It is clear from the figure that the kernel-based solution performs the worst. It produces synthetic instances that are wildly spread around the manifold; we presented a justification for this in the previous chapter. SMOTE offers a clear improvement to the kernel-based method; however, its failings are still apparent. The most prominent issue in its generation is that it synthesizes some points far from the manifold, and in general, has a lot of variance orthogonal to the manifold. In addition, SMOTE produces dense clusters of instances along the manifold and leaves vast empty spaces between them.

As we expect, the manifold-based methods are good at synthesizing instances along the manifold. The data that they synthesize have slightly different properties due to the difference in biases. Visually, denoising autoencoder generates a sprinkling of instances along the manifold, much like a Canadian snowplow travelling down the road, spreading salt crystals after an ice-storm. PCA has more orthogonal variance relative to DAE. Continuing with the metaphor, we can imagine PCA spreading the salt, with some of it bouncing off the roadway and onto the shoulder.

4.5.3 Swiss Roll

Like the helix, the swiss roll is a common dataset in the manifold learning literature [Silva and Tenenbaum, 2002, Weinberger et al., 2004, van der Maaten et al., 2009, Sparks and Madabhushi, 2013, Jia Wei et al., 2008]. Its shape is reminiscent of the Central European pastry. The swiss roll is a plane defined in a 3-dimensional space as $x_1 = y_1 cos y_1$; $x_2 = y_1 sin(y_1)$; $x_3 = y_2$; $y_1 \in [\frac{3\pi}{2}, \frac{9\pi}{2}]$; $y_2 \in [0, 15]$. The training and synthetic swiss roll data is presented in Figure 4.7; the kernel-based method has been omitted due to its poor results. The training instances are plotted as small turquoise (light grey) circles and the synthetic instances are the larger orange circles (dark grey).

The advantage of the manifold-based methods is once again very clear here. Specifically, SMOTE synthesizes points in the vast vacant regions that are not populated by the swiss roll. Three instances are, for example, synthesized in the void that is the centre of the swiss roll. In addition, many synthetic instances span the empty region between the inner and outer layer of the swiss roll. In both cases, SMOTE is clearly placing synthetic instances in regions of the dataspace that are not part of the target distribution. Alternatively, the instances synthesized with the manifold-based synthetic oversampling framework via PCA and the denoising autoencoder stay within the regions reasonably occupied by the swiss roll distribution.

When we take a macro-scale view of the synthesized data, we see that instead of

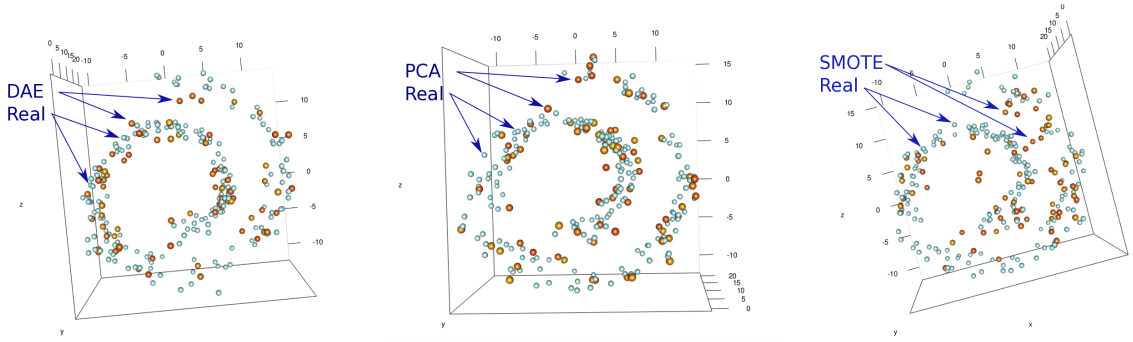


Figure 4.7: From left to right, swiss roll data synthesized by denoising autoencoder (DAE), PCA SMOTE-based methods.

a swiss roll, SMOTE has produced two clusters of synthetic instances. It generated a small cluster on the left of the dataspace. This cluster represents that small area of the swiss roll. The other cluster is much larger in terms of area of population. It occupies much of the central and upper left region of the plot, and sparsely covers both the swiss roll and the void spaces in-between the layers of the roll. The manifold-based synthetic oversampling framework, however, synthesizes instances along the manifold. Therefore, we see that the synthetic instances are sprinkled over all regions of the swiss roll in a manner that is consistent with the target distribution.

This demonstrates, once again, that SMOTE's means of synthesizing data is inappropriate for data that conforms to the manifold property. In particular, we see that synthetic instances are spread orthogonally from the manifold into low probability regions of the dataspace. Behaviour of this nature could significantly impact the performance of the induced classifier if the orthogonal spread produced by SMOTE reaches into the majority class. Give the spread that is visible in each of our examples thus far, the negative impact is a clear risk.

With respect to the manifold-based synthetic oversampling framework, we see that the two implementations of the framework have minimal orthogonal spread. Thus, with manifold-based synthetic oversampling, there is less risk of negatively impacting classification by erroneously added synthetic minority instances into the majority space. Moreover, the framework induces a model that facilitates the synthesization of points that spread along the manifold. This provides the potential for increasing representation in the sparse regions of the target distribution.

4.6 Experimental Set Up

We present two classes of experiments in order to demonstrate the relative strength of our proposed framework in the context of class imbalance and synthetic oversampling. The first set of experiments are conducted on real-world gamma-ray spectral datasets and the second set of experiments utilize the manifold-augmented UCI data proposed in the previous chapter.

Once again, we use the manifold-augmented UCI data because this enables us to control for all of the other complicating factors, such as class overlap, subconcepts, *etc.*, that are implicit in the dataset. Using this method, we get a baseline AUC performance prior to augmentation, and we are able to examine how the performance changes as manifold conformance is increased.

4.6.1 Methodology

The methodology utilized for the UCI data is exactly the same as that discussed in the previous chapter. In particular, we have repeat the previous experiments that demonstrated that the existing synthetic oversampling methods perform poorly on data that conforms to the manifold assumption with PCA and the denoising autoencoder formalizations of our proposed framework. Once again, we evaluate the systems by examining the reduction in the AUC between the baseline dataset and the manifold-augmented versions of the dataset (the loss). This process holds the other properties of each domain constant, and thus, any decrease in the AUC after manifold-augmentation can be attributed to the presence of the manifold. Therefore, a synthetic oversampling method that is robust on manifold data should have a small loss value.

We compare the existing state-of-the-art synthetic oversampling methods to our proposed framework based on the mean loss values incurred by the five classifiers implemented in Weka [Hall et al., 2009] (SVM, MLP, J48, kNN, NB) after synthetic oversampling is applied. The mean scores are calculated over 30 repeated trials with 25 minority training instances sampled without replacement for synthetic oversampling during each trial. For further details of the methodology, the reader is directed to the previous chapter. In addition to the loss score, we compare the methods based on the AUCs produced for $p = 0$ and $p = \{0, .., 90\}$.

Given the large data size of the gamma-ray spectral datasets, and our specific interest in the overall performance impact of each method on the baseline classifier, we report the mean AUC and standard deviation recorded over 5×2-fold cross validation. Although

10-fold cross validation is typical, it has been observed that 5×2-fold cross validation has a lower probability of issuing a Type I error as compared to k-fold CV [Dietterich, 1998]. Furthermore, the large size of the gamma-ray spectral datasets do not necessitate 10-fold cross validation.

4.6.2 Data

Gamma-Ray Spectral Data

Gamma-ray spectral data are collected and analyzed for a wide variety of important experimental and practical purposes, such as isotope classification in the lab [Olmos et al., 1991], the analysis of mining ore [Yoshida et al., 2002], monitoring of ports of entry for the importation of illicit nuclear material [Kangas et al., 2008] and the policing of the comprehensive nuclear test-ban-treaty [Bellinger and Japkowicz,]. In this work, two classes of gamma-ray spectral data that were collected and analyzed by the Radiation Protection Bureau at Health Canada are considered.

Health Canada is a federal department with the mandate to assist Canadians in maintaining and improving their health[1]. The Radiation Protection Bureau operates within Health Canada's mandate with the purpose of promoting and protecting the health of Canadians by assessing and managing risks posed by exposure to radiation at home, work and in the broader environment[2]. To this end, the Radiation Protection Bureau has set up radiation monitoring stations at key sites around the country, including in major cities and near nuclear power plants and nuclear industries, such as medical isotope production facilitates.

The existing monitoring system involves a simple threshold on the dose rate, and/or a threshold on regions-of-interest in the spectra, which are associated with particular isotopes. The goal of this process is to monitor and flag isotopes of interest and detect any generally anomalous events. Each spectra that is flagged by the system is then analyzed by a physicist at the Radiation Protection Bureau to determine the source and ensure that the isotopes of interest are being emitted in safe amounts and at an acceptable frequency. The threshold-based system, however, flags a large number of false positives, which leads to a high cost in terms of human analysis and a potential lag in evaluation.

The general work of our lab in conjunction with the Radiation Protection Bureau has been to devise more sophisticated means of anomaly detection and classification

[1]See http://www.hc-sc.gc.ca/ahc-asc/activit/about-apropos/index-eng.php.

[2]See http://www.hc-sc.gc.ca/ahc-asc/branch-dirgen/hecs-dgsesc/sep-psm/rpb-br-eng.php.

of isotopes of interest. The work in this thesis is particularly focused on the task of classifying a general category of isotopes of interest. The complicating property of this data is the degree of imbalance between the background class and the class of isotopes of interest.

In addition to the national monitoring stations, the Radiation Protection Bureau collaborates with various Canadian security agencies, such as The Canadian Nuclear Safety Commission, Defence Research Canada, Canadian Security Intelligence Service, *etc.*, to deployed gamma-ray spectrometers during high profile events. These agencies deployed gamma-ray spectrometers in and around the Greater Vancouver Area during the 2010 Olympics in order to gather and monitor gamma-ray spectral for isotopes of interest that may signify a person transporting a material that poses a radioactive threat to participants and spectators of the Winter Games. Whilst the Games have long since closed, the data is highly imbalanced and provides an excellent platform on which to evaluate our proposed manifold-based synthetic oversampling framework.

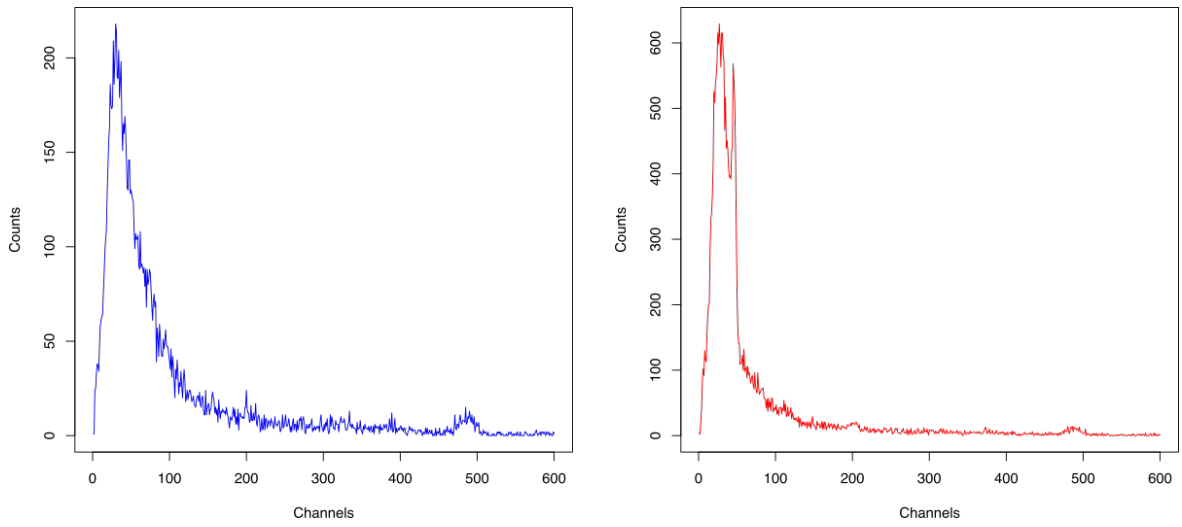
Data Collection: Sodium Iodide detectors are utilized in the national monitoring system and were deployed during the Vancouver 2010 Olympic Games. During the Winter Games, response time was a clear priority, and as such the instruments recorded one measurement per minute; the measurements are recorded as counts per photon energy (keV).

As a result of the one-minute samples, the produced gamma-ray spectra are very noisy. Two examples of gamma-ray spectra collected during the Vancouver 2010 Winter Olympics Games are presented in Figure 4.8 (i) and Figure 4.8 (ii); these correspond to a pure background reading and a background plus Technicium-99, respectively. The latter is from the minority class in our experiments. A distinction can be seen in these two spectral at channels below 100; this is the area of the spectrum affected by Technicium.

In the plot, energy is represented in terms of channels on the x-axis, and the counts, which indicate the intensity, are recorded on the y-axis. Unshielded and poorly shielded isotopes produce specific and identifiable peaks in the background distribution, as the signals are not muted by shielding materials. In the case of Technicium-99, we clearly see this in the low energy range of the spectrum[3]. As the amount of shielding increases and/or the time/distance between the source and the sampling station increases, the representative peak becomes extremely eroded. This is a factor that machine learning, in comparison to the existing system, is particularly well versed to cope with.

In the Vancouver Winter Games dataset, there are $39,000$ background instances and

[3]Technicium-99 is a common medical isotope



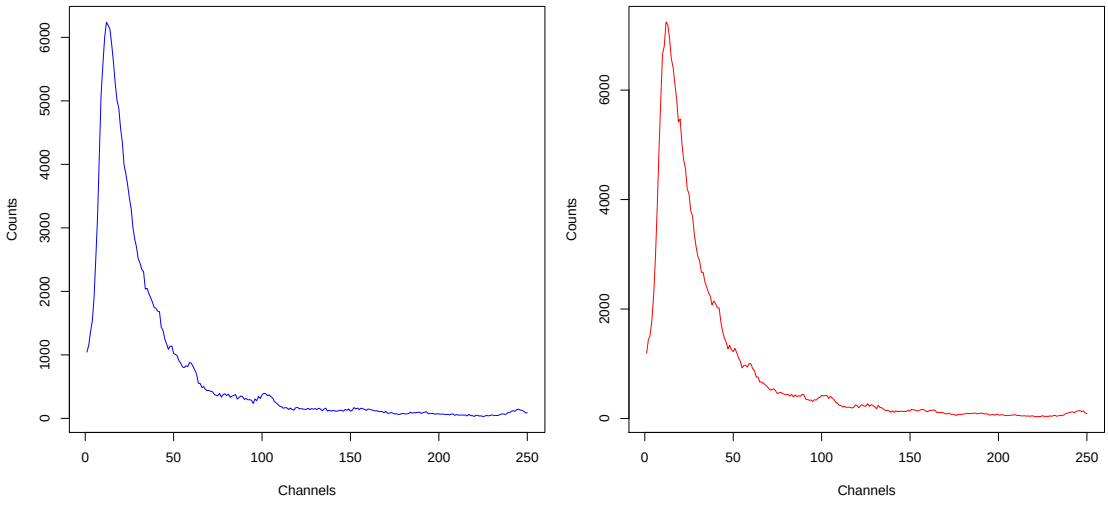
(*i*) (*ii*)

Figure 4.8: Two gamma-ray spectra from the Vancouver dataset. Sub-figure *(i)* depicts a background instance and sub-figure *(ii)* depicts an instance containing Technicium.

39 minority class instances composed of three medical isotopes; namely Iodine, Thallium and Technicium (one of the stations also had readings for Caesium, which were the result of a check-source). Once again, given the high-dimensionality and so few minority training instances, this is a clear example of absolute imbalance.

The environmental monitoring data is collected in fifteen-minute samples by the national monitoring network of gamma-ray spectrometers. The network is designed to detect threats to human health and environment. The vast majority of measurements are solely affected by elements in the local background; these instances are considered to be of no interest. Alternatively, non-background spectra that have been affected by specific isotopes are to be detected and subsequently reviewed by physicists. An example of a background sample from the Thunder Bay monitoring station is displayed in Figure 4.9 (i) and a background sample affected by Technicium is shown in Figure 4.9 (ii). Once again, we can see that Technicium affects the spectrum at channels below 100. It is also clear from these examples that the fifteen-minute samples produced by the national monitoring network have much less noise than those from the Winter Games.

Given the large number of readings resulting from the network each day, machine classification is essential to ensure that the appropriate spectra are given attention in a timely manner. The gamma-ray spectral datasets reported are produced on approxi-



(i) (ii)

Figure 4.9: Two randomly selected examples from the national monitoring network. Sub-figure *(i)* depicts a background instance and sub-figure *(ii)* depicts an instance containing Technicium.

mately four months of sample data. During this period, 11,602 spectra were recorded; 29 of these spectra were affected by technicium. The latter forms the class of interest. Given the very small number of minority class instances and the very high-dimensionality, this is clearly an excellent example of a very challenging and interesting form of absolute imbalance [He and Garcia, 2009]. Each spectra is composed of 250 channels (dimensions).

Data Preprocessing and Profiling: The Radiation Protection Bureau undertook the task of preprocessing the data prior to its utilization in the design and implementation of the machine learning-based system. Subsequently, the data was provided from the sampling stations in timestamped *csv* files, accompanied by corresponding timestamped label files which specify the occurrences of isotopes along with specific isotope names. This labelling was the joint result of their current identification process that involves manual inspection by the analysts.

UCI Data

The sixteen manifold-augmented UCI datasets that were introduced in Chapter 3 are reused in this chapter in order to compare the proposed synthetic oversampling framework to the state-of-the-art methods analyzed in the previous chapter. Please refer to the previous chapter for further details regarding the specific datasets.

4.7 Experimental Results

4.7.1 Gamma-Ray Spectra

The mean results of the five times two-fold cross validation experiments on the Saanich, Thunder Bay and Olympics GRS dataset are presented in Table 4.1. In each case the manifold-based synthetic oversampling framework outperforms the non-manifold-based methods. The columns in Table 4.1 indicate the mean AUCs and standard deviations collectively produced by SVM, MLP, kNN, NB and J48 classifiers on training sets balanced via the corresponding synthetic oversampling method. The AUC is particularly appropriate for domains such as this because it is not affected by class imbalance. The top row includes the baseline results for the set of classifiers.

Finally, we perform statistical analysis via the paired t-test for each gamma-ray spectral dataset with $\alpha = 0.05$ (all of our statistical tests were conducted with this alpha value). Since we are comparing the manifold-based synthetic oversampling framework to

Table 4.1: The mean 5x2CV AUC results for each method on the GRS dataset.

	Saanich		Thunder Bay		Olympics	
	Mean	sd	Mean	sd	Mean	sd
Baseline	0.730	0.046	0.862	0.008	0.755	0.069
Manifold-based	**0.894**	0.049	**0.943**	0.011	**0.811**	0.026
SMOTE K3	0.878	0.007	0.878	0.008	0.777	0.081
SMOTE K5	0.866	0.009	0.866	0.009	0.691	0.094
SMOTE K7	0.834	0.021	0.832	0.020	0.535	0.091
Tomek K3	0.777	0.065	0.878	0.009	0.775	0.079
Tomek K5	0.716	0.065	0.865	0.010	0.670	0.084
Tomek K7	0.600	0.045	0.824	0.023	0.543	0.029

various parameterizations of SMOTE, with and without the removal of Tomek links, we take just the best parameterization of SMOTE for each dataset in our statistical analysis.

Our statistical analysis shows that whilst the mean AUC of 0.894 that was produced by the proposed manifold-based method on the Saanich dataset is better than the competing systems, which produced a top AUC of 0.878, we cannot reject the null hypothesis for the given α-value. Alternatively, based on our experiments, we have evidence to reject the null hypothesis with $\alpha = 0.05$ for both the Vancouver and the Thunder Bay datasets. Thus, we find that the mean AUCs of the proposed system is superior on all three gamma-rays spectral datasets and that it is statistically significantly better on two of the three.

4.7.2 UCI Data

Analysis of Loss

Table 4.2 displays the best loss values for the manifold-based systems and the non-manifold-based systems on the 16 UCI datasets. Specifically, for each dataset, we selected the loss score for the best manifold-based method (PCA or DAE) and the best non-manifold-based method (SMOTE, Tomek, borderline SMOTE); the latter results were first reported in the previous chapter.

The ideal loss value is zero, indicating that the mean AUC produced by the system did not degrade as a result of the added manifold. Our hypothesis states that manifold-based

Table 4.2: This table presents the degradation between $p = 0$ and $p = 90$ of the classifiers after the application of manifold-based synthetic oversampling and non-manifold-based oversampling.

Dataset	Best Manifold-Based	Best Non-Manifold-Based
Letter	0.078	0.116
Musk2	0.018	0.133
Opt Digits	0.001	0.014
Ozone 1hr	0.051	0.055
Pima	0.018	0.028
Sonar	0.061	0.074
Vehicle	0.056	0.062
Wave Form	0.038	0.074
Yeast	0.018	0.063
Satlog	0.067	0.085
Breast	0.002	0.001
Ecoli	0.016	0.022
Heart-Statlog	0.001	0.008
Ionosphere	0.055	0.033
Pen Digits	0.016	0.031
Segment	0.035	0.040

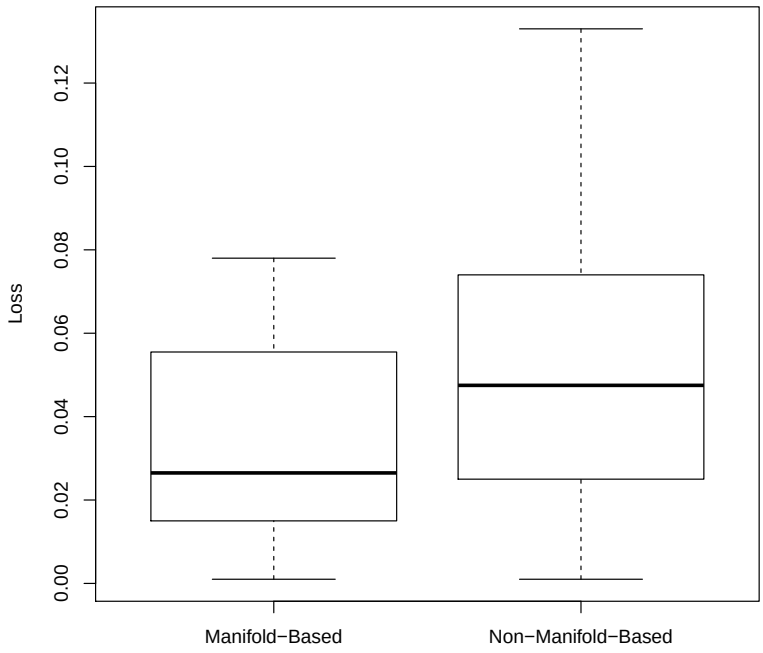


Figure 4.10: Boxplot of loss scores for the best manifold vs non-manifold based systems.

methods should be superior according to this metric. Indeed, this is the case suggested by these UCI experiments. Fourteen of the sixteen datasets have lower loss scores when the manifold based system is applied; these are highlighted in grey. To further emphasize the general superiority of the manifold-based system, Figure 4.10 depicts the box and whisker plots for the manifold-based and non-manifold-based systems. This illustrates that the majority of losses for the manifold-based system are below the median loss of the non-manifold-based system. This is indicative of the significance of the competitive advantage offered by our manifold-based approach on data that conforms to the manifold assumption.

AUC Results

Of what value is a classification system that is robust to manifold data, but generally produces low AUCs? Robustness to the manifold, as demonstrated above, is of great importance; however, it cannot be our only consideration. The synthesized points must

Table 4.3: This table presents total number of AUC wins for each synthetic oversampling system of the UCI data with $p = 0$ and based on the average over all p-values from 0 to 90.

	Wins	
Dataset	$p = 0$	mean($p = \{0, .., 90\}$)
SMOTE	1	0
Tomek	3.5	0
PCA	7	3
DAE	4.5	13

not cause a significant degradation on manifold data, and they must lead to an improvement over the baseline classifier. For this reason, we study the relative performance of each system according to the AUC.

Table 4.3 presents an aggregation of the top rankings of each synthetic oversampling system on the UCI datasets. The first column ($p = 0$) refers to the UCI datasets in their original forms. In the case of a tie between two methods, 0.5 is attributed to each. The manifold-based methods were the top ranking system $7 + 4 = 11$ times out of 16 and tied once with a non-manifold-based method for $p = 0$. In direct comparison with the mean of the baseline classifiers, SMOTE is only superior on 8 of the 16 datasets, whereas the manifold-based approach is better 12 out of 16 times. Thus, both classes of system generally produce an improvement over the baseline classifiers, with our proposed manifold-base approach affecting a greater increase in performance.

The second column of Table 4.3 reports the total number of wins when the performance is averaged across the UCI datasets with manifold augmentation from $p = 0$ to $p = 90$. This column emphasizes the general superiority of the manifold-based methods when the manifold property exists. The manifold-based approach is always better on the datasets that have been manifold-augmented.

Finally, we gauge the statistical significance of the difference in performance using the t-test. The manifold-based system and non-manifold-based system that produced the best loss on each dataset is selected and their performance over the 30 trials with $p = 0$ and $p = 90$ is evaluated. In the case of $p = 0$ the manifold-based method is statistically significantly better 6 times and a non-manifold method is significantly

better twice. On the datasets that conform to the manifold assumption more strongly ($p = 90$), a manifold-based method is significantly better 11 times.

4.8 Discussion

Gamma-Ray Spectral Classification

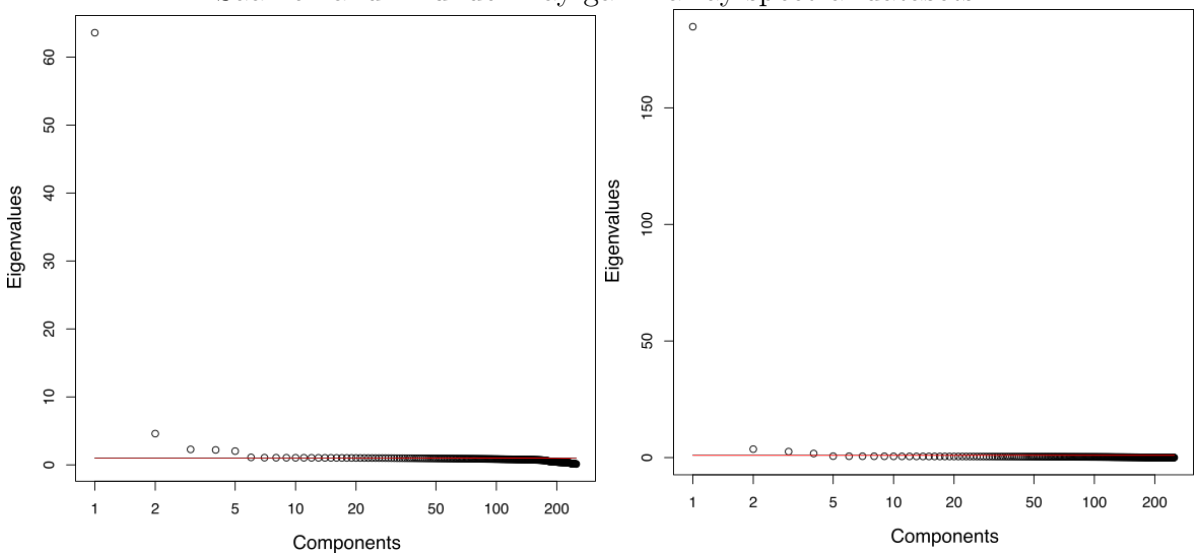
These results show that the manifold-based framework is superior on our three real-world, high-dimensional datasets, that are believed to conform to the manifold assumption. This provides evidence to confirm our hypothesis that manifold-based synthetic oversampling is the most appropriate choice for managing imbalanced classification tasks that conform to the manifold assumption.

To validate conformance to the manifold assumption, we use the factor analysis of the principle components and the scree test as described in the pervious chapter. The factor analysis is presented Figure 4.11 for the three gamma-ray spectral datasets and the UCI wave dataset. We use the UCI wave dataset for comparison because it has the smallest $M(\cdot)$ score amongst the UCI dataset. Each plot depicts the eigenvalues for the corresponding dataset on the y-axis and the sorted principle components in log-scale for ease of visualization on the x-axis. We also include a red solid line at $y = 1$ in each figure for reference. The line at $y = 1$ does not have a particular significance beyond providing a reference point for the comparison between graphs.

The factor analysis figure shows that the first few principle components of the gamma-ray datasets have much larger eigenvalues values than the UCI wave dataset and that a much smaller portion of the eigenvalues have large values. The UCI wave dataset has 13 out of 40 eigenvalues greater than 1, whereas the Saanich dataset, for example, has 6 out of 250. This trend is similar for each of the gamma-ray spectral datasets. Thus, we can infer that the gamma-ray spectral datasets have very strong conformance to the manifold assumption because a very small number of their components, in relative terms, hold most of the variance. We can make a similar claim regarding the wave dataset since most of the spread is in less than half of the principle components, but it is not as strong of a conformance to the manifold assumption.

The scree test is an alternative way of examining conformance to the manifold assumption. It shows that the implicit dimensionalities of the gamma-ray spectral datasets are much lower than the physical dimensionality. The $M(\cdot)$ score are significantly smaller than any in our UCI experiments and the relative performance results are very favorable

Saanich and Thunder Bay gamma-ray spectral datasets.



Vancouver gamma-ray spectral dataset and UCI wave dataset.

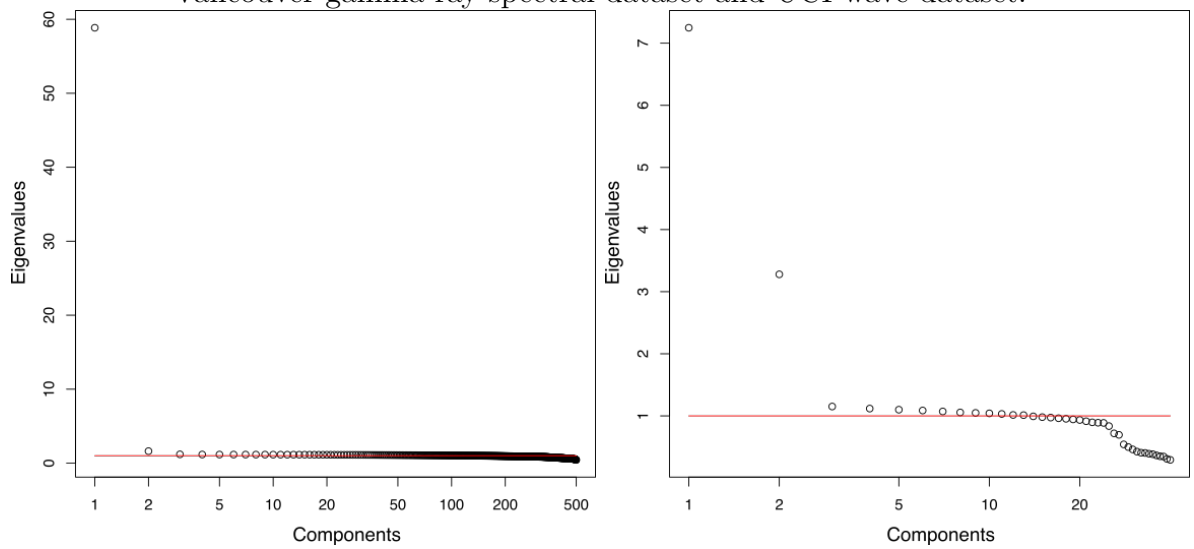


Figure 4.11: This figure presents the factor analysis plots for each gamma-ray spectral dataset and the UCI wave dataset.

for our system. The $M(\cdot)$ scores for each UCI dataset and three gamma-ray spectral datasets are displayed in Figure 4.4. The datasets are sorted by the conformance score so that the datasets appearing at the top conform the most to the manifold assumption. Indeed, the three gamma-rays spectral datasets conform most strongly to the manifold assumption according to the $M(\cdot)$ score.

This illustrates the practical importance of utilizing knowledge regarding the domain for synthetically oversampling the data. We see that on these three real-world classification tasks, not utilizing the knowledge regarding the presence of the manifold places a lower ceiling on the potential of synthetic oversampling.

Table 4.4: Manifold conformance score based on $M(\cdot)$ for each test dataset.

Dataset	$M(\cdot)$
GRS-Vancouver	0.004
GRS-Thunderbay	0.016
GRS-Saanich	0.02
Wave-form	0.025
Ozone One	0.083
Sonar	0.100
Segment	0.105
Musk	0.108
Vehicle	0.111
Satlog	0.139
Opt Digits	0.172
Letter	0.188
Ionospher	0.206
Breast	0.222
Pen Digits	0.250
Diabetes	0.375
Heart	0.385
Ecoli	0.571
Yeast	0.625

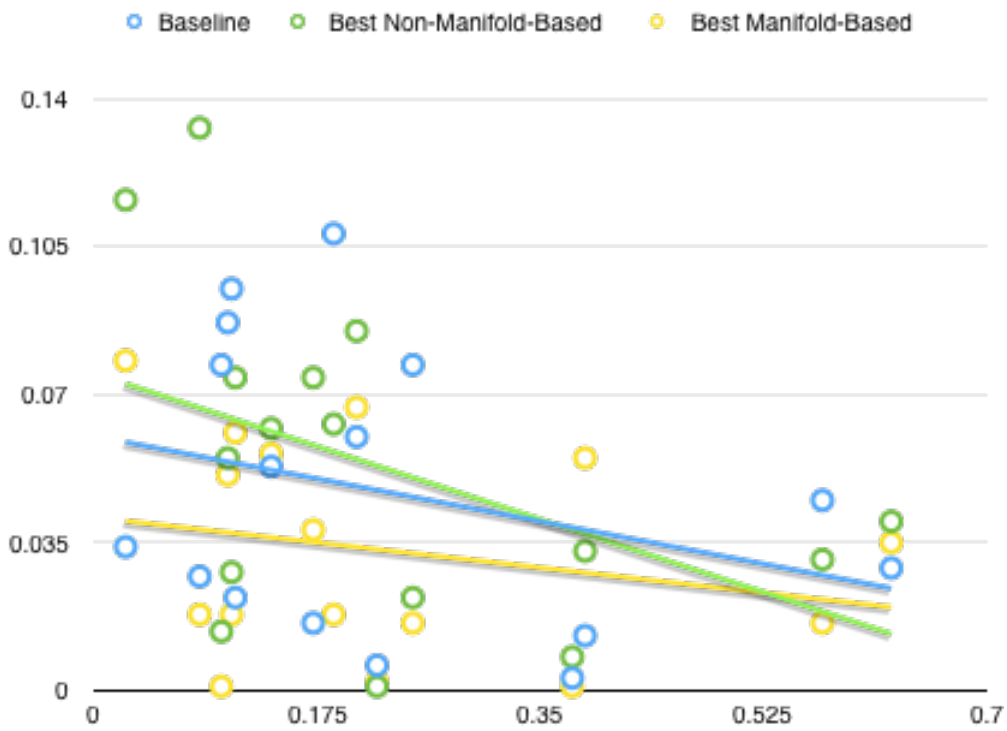


Figure 4.12: Comparison of $loss(D)$ versus $M(D)$ for each method on the UCI datasets.

UCI Datasets

The purpose of our empirical analysis of the UCI domains is to understand the general affect of the manifold property on the state-of-the-art oversampling methods, and to evaluate our hypothesis that manifold-based oversampling methods are more robust data synthesizers when the manifold assumption holds.

The UCI repository provides access to a great number of test domains on which to conduct our general analysis, and by performing manifold-augmentation on these datasets, we isolate the impact of the manifold on the synthesization of data for the purpose of managing class imbalance. This is done by examining the AUC performance before and after augmentation.

The results based on this evaluation strategy clearly demonstrate that our proposed manifold-based synthetic oversampling framework has a performance advantage over the existing methods when evaluated based on its effect on the AUC and in terms of the degradation in the AUC caused by the presence of the manifold property in the data.

Figure 4.12 emphasizes the significant advantage that manifold-based synthetic oversampling has in comparison to synthetically oversampling with SMOTE. The $M(D)$

score is plotted on the x-axis and the $loss(D)$ is plotted on the y-axis. Smaller $M(D)$ values indicate that the dataset D has a stronger conformance to the manifold property, whereas a lower $loss(D)$ scores indicates that the method is less impacted by the manifold. Therefore, a point in the top-left indicates that the dataset has a strong conformance to the manifold property and that the method applied was significantly impacted by the manifold, whereas a point in the bottom-right indicates that the data has a weak conformance to the manifold property and that the method was minimally impacted by the manifold. The key point is that a robust method will have a low $loss(D)$ regardless of the $M(D)$.

In the plot, each blue point indicates the baseline performance on a UCI dataset, and the blue line is a linear regression line fit to the points. Likewise, the green items correspond to the non-manifold based method (SMOTE) and the yellow items to the manifold-based synthetic oversampling approach. The yellow regression line for manifold-based synthetic oversampling is below that of the baseline results and flatter than the baseline regression line. This shows that the classifier is less affected by the manifold after synthetic oversampling than before. Alternatively, the regression line for SMOTE is above the baseline for $M() < 0.35$, which indicates that SMOTE causes additional degradation in performance on domains that conform to the manifold property.

In addition to general robustness on manifold data, a useful tool for synthetic oversampling must also produce improvements in AUC relative to the competing methods. Our results show that, not only is our proposed method robust in terms of the manifold, it also improves the AUC more than the SMOTE-based alternatives do. On 15 of the 16 manifold-augmented datasets with $p = 90$, manifold-based oversampling produces better AUCs. Eleven of these cases are statistically significant improvements for $\alpha = 0.5$. Thus, there is very good justification for utilizing the manifold-based framework for synthetic oversampling.

A very interesting property that we identified during our analysis of the manifold-based methods is that the PCA realization of the framework performed better when the manifold was less prominent, whereas the denoising autoencoder was clearly stronger when the manifold was more prominent. Our belief is that our proposed PCA sampling algorithm implicitly causes more spread orthogonal to the manifold, and thus, works well when the manifold is not as tightly bound. This is also evident in the visualization of the synthesized helix and swiss roll data.

Parameter selection is generally as important as it is challenging in machine learning; there is no exception for synthetic oversampling. In cases of rarity, as we have in abso-

lute imbalance classification, parameter selection is, in fact, more important and more challenging. We must select good parameters so the performance of the system does not degrade during application and we must do it with few examples. In most cases of class imbalance there is not enough data to independently select the parameters of the synthetic oversampling system. Moreover, many of the existing approaches (most notably the class of SMOTE-base systems) do not provide a means by which to select parameters independently of the classifiers themselves.

The typical strategy in the literature for selecting the k parameter for SMOTE is either not specified, or is to run cross-validated classification experiments on the data with using different k values for SMOTE [Chawla et al., 2002]. The best system, specifically which k to use, is decided after peeking at all the results. This methodology is only appropriate in an academic setting since there is no way to "peek" in the real-world. For lack of a better alternative, however, this is the method that we applied for SMOTE.

The denoising autoencoder formalization of our framework provides a very practical means of parameter selection without peeking at the results. This is the method that we have followed throughout, have still managed to outperform SMOTE! Specifically, we perform a random search of the denoising autoencoder parameter space, evaluating each model based on the reconstruction error produced on the minority training set (as described in Algorithm 3). We keep the model with the smallest reconstruction error, and thus, there is no need to peek at the classification results. A single model is selected before any classification is performed. The ability to run parameter selection for the synthetic oversampler without the requirement of an independent and labelled set of instances that are separate from the training set provides a significant practical advantage over the existing methods!

4.9 Conclusion and Future Work

The state-of-the-art methods in synthetic oversampling eschew the no-free-lunch theorem and propose solutions that naïvely ignore the underlying data properties. Whilst in some cases we do not have access to insight regarding the latent characteristics of the data, our collective metamorphosis into data scientists has put us in close collaborative positions that often enable the use of beneficial data properties. In the previous chapter, for example, we described our process of identifying soft conformance to the manifold assumptions in our target gamma-ray spectral data. Moreover, our years of collective experience have provided empirical evidence to emerge around the properties of certain

data domains. The end result is that we often do have specific information about the properties of the data that we can harness in the emergent solutions.

Our research focuses on knowledge of the target data's conformance to the manifold assumption. In the previous chapter, we demonstrated that the existing methods of synthetically oversampling cause a degradation in performance relative to the baseline degradation when the manifold assumption holds. In these cases, we argued, a manifold-based method of synthetically oversampling is required.

In this chapter we address the identified weakness in the state-of-the-art approaches to synthetically oversampling by proposing a manifold-based framework for synthetically oversampling. In constructing the framework, we are recognize the depth of current research into manifold learning and acknowledge that the biases and assumptions that are implicit in these methods render them more or less suitable for the surfeit of manifold learning tasks. In addition to proposing the framework, we have demonstrated two formalizations of it and demonstrate that the utilization of the framework on data that conforms to the manifold assumption results in less loss and produces superior AUC performance.

We demonstrated that the denoising autoencoder is a particularly good formalization of the framework; however, many interesting questions remain in regards to this novel use of the denoising autoencoder. In our future work, we will study application appropriate metrics for model selection. Whilst the standard reconstruction error worked well in our study, we believe that an even better, application specific method for model selection exists. In addition to PCA and DAE, we have mentioned that there is a significant breadth of solid research into manifold learning. Our ongoing work continues to explore and understand the properties and strengths of these methods for synthetic oversampling. This includes our consideration of a means by which to select the most appropriate manifold learning technique for a given dataset via meta-learning.

With respect to manifold learning algorithms, we are interested in exploring the impact of training set size of local neighbourhood-based manifold learning, such as local linear embedding versus more global approaches like autoencoders and PCA. Moreover, we are interested in exploring the relationship between data classes and manifold learning, with a focus on how to synthesize instances from regions of the manifold that will be most effective from a synthetic oversampling perspective. We have theories on how to do this for the denoising autoencoder formalization of the framework, but we would like to generalize it.

Finally, an important, but more general, next step in our research is to explore how

best to apply our framework to multi-class classification tasks and analyze the benefit of doing so.

Chapter 5

Conclusion

5.1 Summary of Work Completed

The results of our efforts to understand the impact of the manifold property on synthetic oversampling and the derivation of a method that is appropriate for synthetically oversampling such data is best emphasized on a chapter-by-chapter basis.

Chapter 1: We began this manuscript by highlighting the benefits of developing machine learning algorithms that incorporate knowledge about the properties of the target data whenever possible and emphasized that, thanks to the increasing trend of collaboration with domain experts, and our growing collective experience in the field, we are better prepared to access this information than ever before. We presented our motivation which focuses the thesis on devising a synthetic oversampling method that is robust on data that conforms to the manifold property, and articulated how we were inspired by our work with imbalanced gamma-ray spectral classification.

Chapter 2: In this chapter we surveyed the field of class imbalance. We emphasized that, although many approaches have been proposed for managing the ill-effects of class imbalance, when the minority training set is very small, the only effective solution is to get more data. The realities of imbalance, however, dictate that additional real training instances are unavailable due to time and/or cost constraints, or the prior class probability, etc. As a result, synthetic instances must be taken in place of real training instances. We continued from here to examine the state-of-the-art in synthetic oversampling and found that none of the existing methods appeared to be appropriate for data that conforms to the manifold property due to their implicit or explicit assumptions. In addition, we surveyed the current state-of-the-art in manifold learning as it has been applied to

machine learning and considered the properties of manifold learning algorithms that are of benefit to synthetic oversampling.

Chapter 3: Having found that no method in literature seemed to be appropriate for synthetically oversampling data that conforms to the manifold assumption, in this chapter we tested our hypothesis that, if the algorithm is not designed for data that contains the manifold property, then it is likely that the method will not be robust when the manifold property exists. We demonstrated how to augment benchmark UCI datasets in order to independently test the impact of the manifold property on existing synthetic oversampling methods, and how to estimate the significance of conformance to the manifold assumption using the scree test. Our results showed that as the conformance to the manifold assumption is increased, the existing methods are negatively impacted by the manifold property when performance is measured in terms of the degradation in the AUC.

Chapter 4: In this chapter, we addressed the failings of the state-of-the-art methods in synthetic oversampling with respect to data that conforms to the manifold property. In particular, we proposed a framework for synthetically oversampling data that conforms to the manifold property. We are particularly excited about the development of this framework as it will allow future practitioners to harness the great breadth of research in manifold learning by selecting the approach that includes the biases and assumptions that are most appropriate for their particular problem. We demonstrated two formalizations of the framework, one based on PCA and the other based on denoising autoencoders, and in doing so, we showed that the proposed framework is superior to existing methods when the manifold property exists in the data.

5.2 Future Work

Class imbalance, synthetic oversampling and manifold learning are diverse, complex and incredibly interesting fields of study in their own right. With this in mind, it comes as no surprise that, in spite of the significant foundation upon which we have set this burgeoning field of synthetically oversampling the manifold, there are many interesting questions that remain to be studied.

- **Chapter 3:** Although we demonstrated the impact of the latent manifold on the existing synthetic oversampling methods in general, manifolds are complex structures and can take a variety of forms. A major future work is to discover and

attempt to categorize the different properties of manifold data in a machine learning context and to further understand the relationship between these properties and synthetic oversampling methods.

- **Chapter 4:** Given the prominence of multi-class classification, and the fact that many of these involve class imbalance, an important next step for this research is to study how best to apply the framework to multi-class problems. In addition, deep learning algorithms have been extremely successful in a wide variety of domains. These algorithms, however, require large datasets for training, which limits their applicability. We see manifold-based synthetic oversampling as having the potential to generate additional training instances for deep learning thereby broadening the scope of applicable domains.

  In constructing the framework, we recognized the depth of current research into manifold learning and acknowledged that the biases and assumptions that are implicit in these methods render them more or less suitable for the surfeit of manifold learning tasks. An important future work involves understanding which properties make a manifold learning algorithm ideal for inclusion in the framework, and performing meta-learning in order to facilitate the suggestion of a specific manifold learning algorithm for a given target dataset. In addition, we continue to consider a general form of model selection for the framework. The reconstruction error performed well for the denoising autoencoder formalization, we believe that the derivation of a synthetic oversampling specific metric for model selection would be ideal.

  We are specifically interested in studying the effectiveness of local neighbourhood-based manifold learning on small training sets. Are, as we suspect, the more globally-based methods more effective in our application, and can the trade-off be mathematically formulated? Generating samples from a learnt model of the latent manifold is a novel and fascinating area for future research resulting from this work. Taken in the context of class imbalance, our objective is to advance our understanding of the data classes and the manifold, and devise means of sampling directly from regions of the manifold that will be most impactful for ameliorating the negative impact of class imbalance. We currently have notions regarding how best to do this for the denoising autoencoder formalization, and continue to test this understanding and extend it to alternate methods.

5.3 Conclusion

The metamorphosis of machine learning from its isolation in sterile labs and academic conferences into the integrated and impactful field of data science has had broad ranging implications. Among these, we can certainly include new found prominence of the wizards of learning algorithms, such as Geoffrey Hinton and Andrew Ng, in popular science and popular media [Hernandez, 2014, Allen, 2015, Buyting, 2015, Simonite, 2014, Hof, 2014, Wong, 2014]. Even more impactful than our increased profile in the social consciousness is the increased collaboration between designers and practitioners of learning algorithms with the domain experts that understand the data and its implicit properties, much like a NASCAR driver understands the subtleties of their home track. The increased collaboration can be seen in the business community with increasing number of jobs, and with data being referred to as the new oil [Dillow, 2013, Kosner, 2014]. In line with this, the Economist magazine is organizing a conference bringing together leading academic thinkers and business people to discuss all aspects of the digital transformation [Eco, 2016]. Likewise, we have seen a steady increase in the impact factor of domain specific journals. The journal Artificial Intelligence in Medicine published by Elsevier, for example, has seen its impact factor increase from 0.672 to 2.016 between 1994 and 2014 along with a significant increase its cumulative citation count [1] [Els, 2014].

Our new-found relationship with domain expertise promotes the effective design of intelligent algorithms. Specifically, we now have greater access to insights regarding the data for which we develop learning algorithms. Thus, we are now, more than ever, in a position to embrace Wolpert's classical no-free-lunch theorem of statistical inference by designing sophisticated machine learning solutions with the properties of the data in mind. To ignore this information is to place an unnecessarily low ceiling on our performance.

In this thesis, we have dealt with synthetic oversampling for class imbalance. We have argued that, in general, it is wise to design machine learning systems that account for knowledge of the data domain and the properties of the data whenever possible. Moreover, we have emphasized that, due to the fact that synthetic oversampling methods are required to make large generative leaps from small training sets, there is a significant benefit from selecting algorithms with biases that match well with the propreties of the target data.

[1] See Microsoft's citation count at http://academic.research.microsoft.com/Journal/244/artmed-artificial-intelligence-in-medicine.

Given our research into the imbalanced classification of gamma-ray spectra with the Radiation Protection Bureau at Health Canada, our specific interest commenced with an examination of how to appropriately synthesize the minority class in the gamma-ray classification problem. This study lead us to understand the negative impacts of conformance to the manifold property on the state-of-the-art in synthetic oversampling methods, and to seek a superior alternative. In this thesis, we have highlighted the impact of the manifold on synthetic oversampling and proposed a framework for synthetically oversampling the manifold. As a result of the proposed framework, we were able to improve the AUC results on all three of our key gamma-ray spectral classification domains and showed that our method is highly beneficial on data that conforms to the manifold assumption in general.

Appendix A

Tables

A.1 Baseline and Synthetic Oversampling Results Table

Dataset	Method	0%	15%	30%	45%	60%	75%	90%	Loss
Breast	SMOTE K3	0.929	0.930	0.935	0.932	0.930	0.929	0.926	0.003
	SMOTE K5	0.930	0.935	0.935	0.934	0.932	0.929	0.931	-0.001
	SMOTE K7	0.934	0.936	0.932	0.932	0.929	0.925	0.925	0.009
	Tomek K3	0.936	0.935	0.937	0.936	0.931	0.929	0.935	0.001
	Tomek K5	0.935	0.939	0.936	0.935	0.934	0.928	0.933	0.002
	Tomek K7	0.936	0.935	0.938	0.938	0.934	0.933	0.932	0.004
	Border K3	0.551	0.553	0.470	0.668	0.518	0.360	0.551	0.000
	Baseline	0.915	0.920	0.920	0.906	0.907	0.912	0.909	0.006
Ecoli	SMOTE K3	0.939	0.930	0.921	0.921	0.917	0.904	0.899	0.04
	SMOTE K5	0.937	0.937	0.926	0.927	0.916	0.908	0.910	0.027
	SMOTE K7	0.941	0.935	0.935	0.928	0.920	0.909	0.904	0.037
	Tomek K3	0.933	0.934	0.928	0.924	0.908	0.906	0.893	0.04
	Tomek K5	0.935	0.938	0.927	0.922	0.916	0.908	0.900	0.035
	Tomek K7	0.937	0.936	0.932	0.924	0.921	0.912	0.904	0.033
	Border K3	0.627	0.644	0.548	0.547	0.517	0.501	0.470	0.157
	Baseline	0.887	0.889	0.865	0.867	0.859	0.859	0.842	0.045
Heart-statlog	SMOTE K3	0.762	0.760	0.763	0.757	0.754	0.749	0.752	0.010
	SMOTE K5	0.761	0.761	0.768	0.764	0.759	0.746	0.753	0.008

Dataset	Method	0%	15%	30%	45%	60%	75%	90%	Loss
	SMOTE K7	0.763	0.765	0.760	0.759	0.756	0.758	0.755	0.008
	Tomek K3	0.764	0.760	0.760	0.759	0.752	0.753	0.749	0.015
	Tomek K5	0.762	0.770	0.764	0.769	0.756	0.757	0.754	0.008
	Tomek K7	0.762	0.765	0.768	0.760	0.744	0.755	0.754	0.008
	Border K3	0.545	0.504	0.437	0.485	0.495	0.562	0.409	0.136
	Baseline	0.755	0.750	0.756	0.751	0.744	0.745	0.742	0.013
Ionosphere	SMOTE K3	0.826	0.819	0.803	0.802	0.798	0.790	0.792	0.034
	SMOTE K5	0.831	0.815	0.810	0.806	0.804	0.797	0.793	0.038
	SMOTE K7	0.830	0.823	0.816	0.807	0.807	0.796	0.797	0.033
	Tomek K3	0.831	0.821	0.803	0.804	0.799	0.794	0.784	0.047
	Tomek K5	0.829	0.821	0.810	0.799	0.795	0.793	0.793	0.036
	Tomek K7	0.832	0.820	0.819	0.808	0.803	0.795	0.794	0.038
	Border K3	0.489	0.482	0.562	0.523	0.574	0.479	0.541	-0.052
	Baseline	0.778	0.766	0.751	0.746	0.738	0.723	0.718	0.060
Letter	SMOTE K3	0.848	0.808	0.766	0.752	0.734	0.719	0.719	0.129
	SMOTE K5	0.845	0.815	0.769	0.752	0.739	0.728	0.723	0.122
	SMOTE K7	0.844	0.820	0.771	0.762	0.736	0.729	0.718	0.126
	Tomek K3	0.851	0.816	0.783	0.762	0.750	0.736	0.729	0.122
	Tomek K5	0.851	0.814	0.782	0.763	0.748	0.730	0.733	0.118
	Tomek K7	0.851	0.825	0.778	0.765	0.752	0.730	0.735	0.116
	Border K3	0.445	0.520	0.470	0.493	0.498	0.482	0.534	-0.089
	Baseline	0.762	0.732	0.691	0.679	0.663	0.664	0.654	0.108
Musk2	SMOTE K3	0.768	0.748	0.725	0.704	0.684	0.658	0.635	0.133
	SMOTE K5	0.776	0.745	0.726	0.700	0.678	0.657	0.641	0.135
	SMOTE K7	0.766	0.756	0.731	0.705	0.681	0.662	0.637	0.129
	Tomek K3	0.773	0.752	0.729	0.709	0.679	0.656	0.640	0.133
	Tomek K5	0.780	0.755	0.730	0.696	0.682	0.655	0.638	0.142
	Tomek K7	0.775	0.748	0.728	0.703	0.681	0.656	0.637	0.138
	Border K3	0.431	0.459	0.511	0.531	0.532	0.501	0.538	-0.107
	Baseline	0.724	0.690	0.668	0.650	0.642	0.630	0.629	0.095
OptDigits	SMOTE K3	0.831	0.828	0.827	0.824	0.818	0.814	0.811	0.020
	SMOTE K5	0.830	0.831	0.827	0.828	0.826	0.820	0.816	0.014
	SMOTE K7	0.837	0.829	0.829	0.828	0.823	0.826	0.823	0.014
	Tomek K3	0.834	0.827	0.827	0.822	0.821	0.812	0.805	0.029

Dataset	Method	0%	15%	30%	45%	60%	75%	90%	Loss
	Tomek K5	0.833	0.834	0.832	0.830	0.825	0.822	0.815	0.018
	Tomek K7	0.836	0.833	0.834	0.827	0.828	0.821	0.818	0.018
	Border K3	0.581	0.546	0.563	0.661	0.562	0.626	0.517	0.064
	Baseline	0.828	0.826	0.819	0.824	0.819	0.820	0.812	0.016
OZoneOnehrCols	SMOTE K3	0.695	0.689	0.680	0.670	0.657	0.644	0.640	0.055
	SMOTE K5	0.710	0.690	0.677	0.667	0.649	0.644	0.638	0.072
	SMOTE K7	0.701	0.698	0.685	0.668	0.657	0.644	0.642	0.059
	Tomek K3	0.711	0.697	0.689	0.676	0.664	0.657	0.647	0.064
	Tomek K5	0.712	0.702	0.693	0.679	0.662	0.655	0.646	0.066
	Tomek K7	0.714	0.694	0.694	0.682	0.663	0.652	0.639	0.075
	Border K3	0.561	0.522	0.540	0.535	0.507	0.523	0.477	0.084
	Baseline	0.625	0.615	0.613	0.603	0.605	0.599	0.598	0.027
Pendigits	SMOTE K3	0.955	0.948	0.944	0.936	0.928	0.934	0.920	0.035
	SMOTE K5	0.957	0.952	0.945	0.942	0.934	0.932	0.922	0.035
	SMOTE K7	0.958	0.953	0.945	0.942	0.930	0.925	0.925	0.033
	Tomek K3	0.960	0.957	0.943	0.939	0.934	0.929	0.922	0.038
	Tomek K5	0.957	0.956	0.943	0.939	0.933	0.929	0.925	0.032
	Tomek K7	0.958	0.950	0.949	0.940	0.935	0.930	0.927	0.031
	Border K3	0.742	0.767	0.774	0.710	0.661	0.801	0.784	-0.042
	Baseline	0.946	0.942	0.928	0.916	0.897	0.884	0.869	0.077
Pima	SMOTE K3	0.638	0.631	0.618	0.611	0.608	0.604	0.603	0.035
	SMOTE K5	0.642	0.620	0.623	0.605	0.609	0.611	0.605	0.037
	SMOTE K7	0.643	0.625	0.628	0.608	0.615	0.608	0.603	0.040
	Tomek K3	0.636	0.630	0.627	0.614	0.614	0.613	0.607	0.029
	Tomek K5	0.642	0.626	0.630	0.616	0.619	0.616	0.614	0.028
	Tomek K7	0.649	0.635	0.633	0.628	0.619	0.619	0.616	0.033
	Border K3	0.525	0.475	0.532	0.500	0.527	0.483	0.512	0.013
	Baseline	0.568	0.565	0.562	0.562	0.559	0.560	0.565	0.003
Segment	SMOTE K3	0.887	0.865	0.852	0.843	0.840	0.831	0.836	0.051
	SMOTE K5	0.889	0.876	0.863	0.856	0.853	0.853	0.843	0.046
	SMOTE K7	0.643	0.625	0.628	0.608	0.615	0.608	0.603	0.040
	Tomek K3	0.895	0.880	0.867	0.856	0.860	0.849	0.845	0.050
	Tomek K5	0.906	0.889	0.872	0.867	0.865	0.866	0.852	0.054
	Tomek K7	0.914	0.887	0.886	0.869	0.872	0.864	0.860	0.054

Dataset	Method	0%	15%	30%	45%	60%	75%	90%	Loss
	Border K3	0.541	0.549	0.486	0.494	0.530	0.474	0.539	0.002
	Baseline	0.864	0.825	0.802	0.794	0.787	0.787	0.777	0.087
Sonar	SMOTE K3	0.736	0.707	0.694	0.683	0.674	0.662	0.654	0.082
	SMOTE K5	0.733	0.701	0.700	0.680	0.675	0.661	0.652	0.081
	SMOTE K7	0.737	0.713	0.700	0.678	0.667	0.662	0.655	0.082
	Tomek K3	0.729	0.714	0.693	0.690	0.665	0.661	0.652	0.077
	Tomek K5	0.726	0.711	0.700	0.680	0.670	0.651	0.652	0.074
	Tomek K7	0.734	0.710	0.686	0.687	0.669	0.664	0.646	0.088
	Border K3	0.496	0.518	0.497	0.508	0.536	0.496	0.511	-0.015
	Baseline	0.724	0.701	0.684	0.675	0.662	0.665	0.647	0.077
Vehicle	SMOTE K3	0.651	0.620	0.609	0.603	0.592	0.582	0.581	0.070
	SMOTE K5	0.657	0.631	0.618	0.603	0.587	0.585	0.583	0.074
	SMOTE K7	0.663	0.631	0.617	0.601	0.593	0.583	0.584	0.079
	Tomek K3	0.656	0.633	0.614	0.603	0.597	0.590	0.594	0.062
	Tomek K5	0.667	0.633	0.621	0.612	0.597	0.589	0.592	0.075
	Tomek K7	0.664	0.641	0.622	0.605	0.603	0.593	0.583	0.081
	Border K3	0.509	0.506	0.501	0.526	0.487	0.514	0.520	-0.011
	Baseline	0.581	0.569	0.566	0.559	0.564	0.557	0.559	0.022
WaveForm-5000	SMOTE K3	0.717	0.699	0.683	0.672	0.659	0.654	0.643	0.074
	SMOTE K5	0.725	0.706	0.688	0.679	0.666	0.655	0.647	0.078
	SMOTE K7	0.728	0.700	0.690	0.677	0.667	0.661	0.650	0.078
	Tomek K3	0.735	0.713	0.695	0.685	0.670	0.658	0.655	0.080
	Tomek K5	0.736	0.714	0.696	0.682	0.676	0.671	0.653	0.083
	Tomek K7	0.735	0.714	0.700	0.683	0.680	0.666	0.655	0.080
	Border K3	0.547	0.512	0.508	0.496	0.508	0.549	0.483	0.064
	Baseline	0.657	0.649	0.641	0.631	0.626	0.624	0.619	0.034
Yeast	SMOTE K3	0.708	0.689	0.678	0.653	0.650	0.635	0.618	0.090
	SMOTE K5	0.703	0.691	0.693	0.659	0.654	0.643	0.637	0.066
	SMOTE K7	0.712	0.698	0.693	0.669	0.655	0.650	0.637	0.075
	Tomek K3	0.705	0.699	0.683	0.660	0.654	0.646	0.630	0.075
	Tomek K5	0.710	0.698	0.693	0.672	0.660	0.651	0.643	0.067
	Tomek K7	0.710	0.705	0.702	0.680	0.661	0.654	0.647	0.063
	Border K3	0.535	0.535	0.514	0.506	0.474	0.496	0.512	0.023

Dataset	Method	0%	15%	30%	45%	60%	75%	90%	Loss
	Baseline	0.602	0.606	0.599	0.587	0.578	0.575	0.573	0.029
Satlog	SMOTE K3	0.770	0.730	0.713	0.710	0.695	0.686	0.677	0.093
	SMOTE K5	0.769	0.742	0.718	0.710	0.692	0.690	0.683	0.086
	SMOTE K7	0.772	0.743	0.719	0.715	0.702	0.695	0.685	0.087
	Tomek K3	0.778	0.741	0.725	0.715	0.701	0.695	0.690	0.088
	Tomek K5	0.781	0.743	0.736	0.714	0.710	0.698	0.696	0.085
	Tomek K7	0.778	0.751	0.731	0.722	0.709	0.697	0.690	0.088
	Border K3	NA	NA	NA	NA	NA	NA	NA	NA[1]
	Baseline	0.675	0.648	0.640	0.635	0.631	0.626	0.622	0.053

[1]Borderline SMOTE failed on this dataset.

Appendix B

Plots

B.1 Baseline Degradation Plot

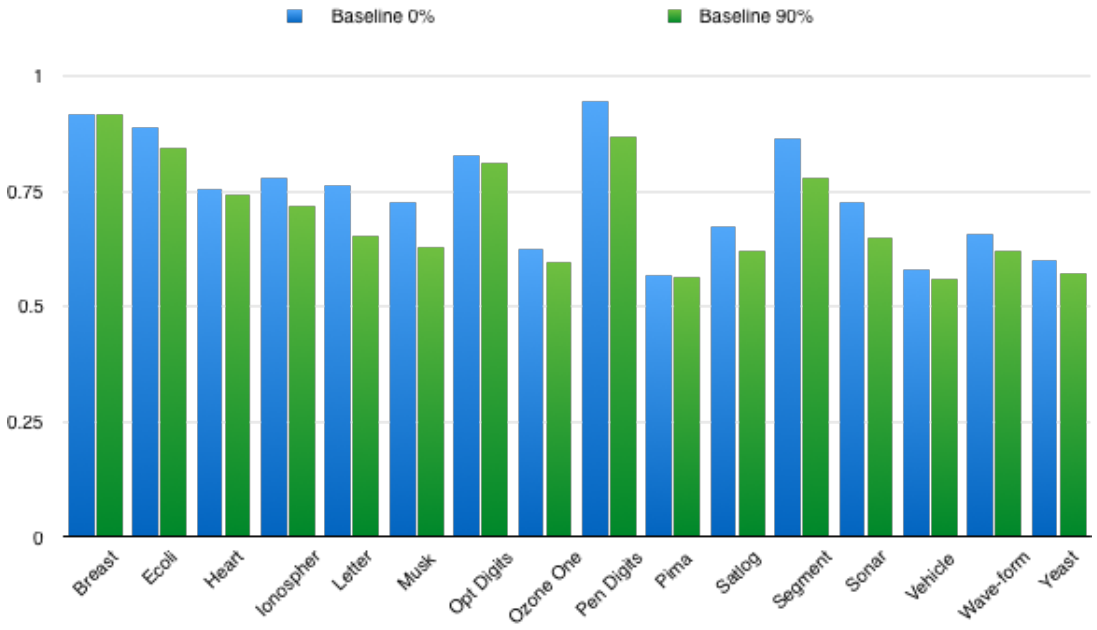


Figure B.1: Degradation of the mean of the baseline classifiers after manifold augmentation on the UCI domains for $p = 0$ and $p = 90$.

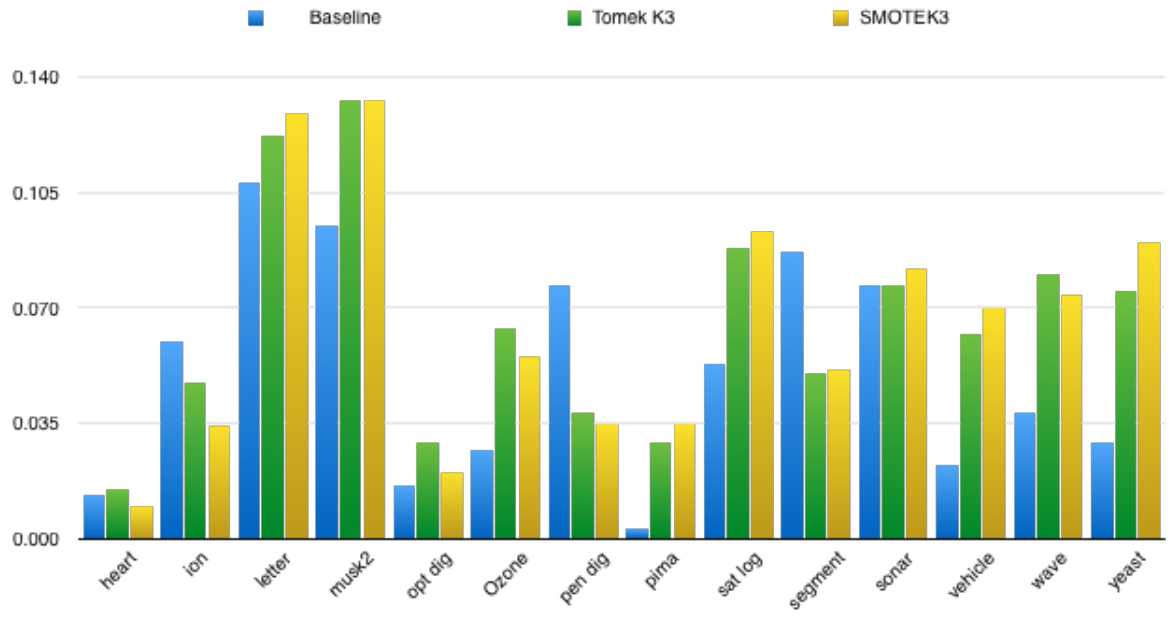


Figure B.2: Comparison of the performance degradation of the mean of the baseline classifiers to SMOTE and SMOTE+Tomek links with K=3 after manifold augmentation on the UCI domains for $p = 0$ and $p = 90$.

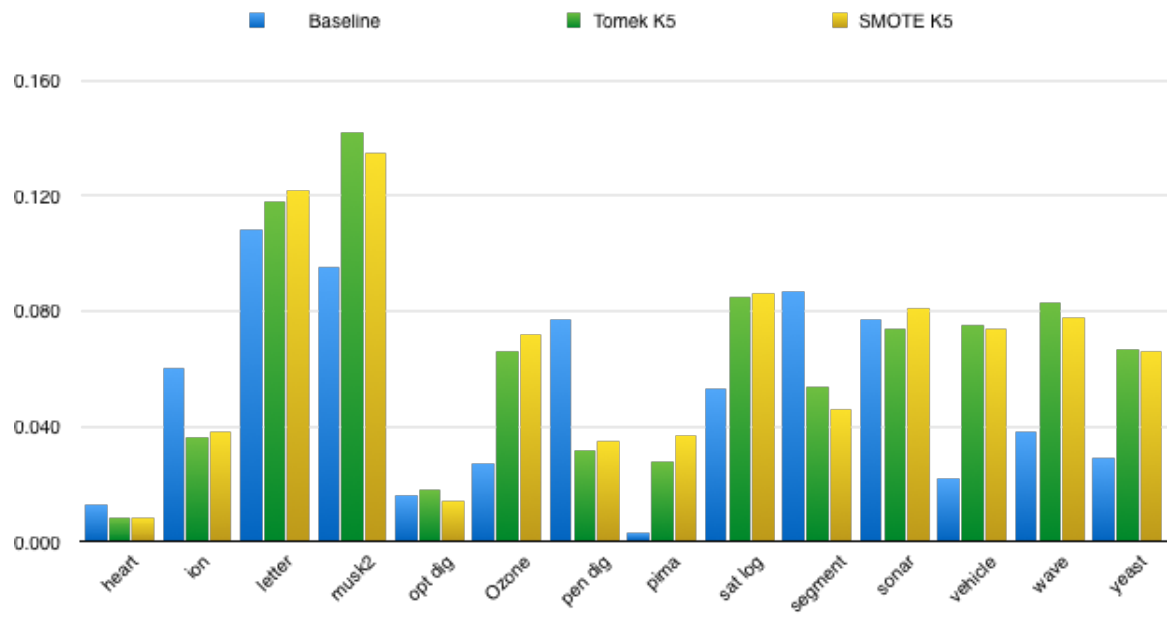


Figure B.3: Comparison of the performance degradation of the mean of the baseline classifiers to SMOTE and SMOTE+Tomek links with K=5 after manifold augmentation on the UCI domains for $p = 0$ and $p = 90$.

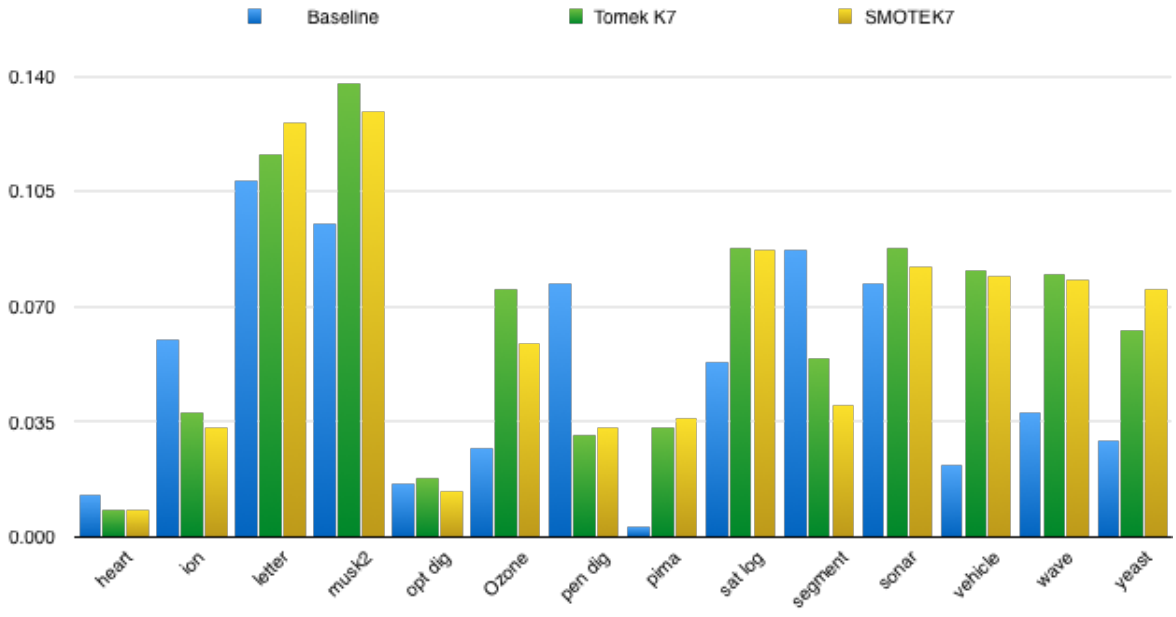


Figure B.4: Comparison of the performance degradation of the mean of the baseline classifiers to SMOTE and SMOTE+Tomek links with K=7 after manifold augmentation on the UCI domains for $p = 0$ and $p = 90$.

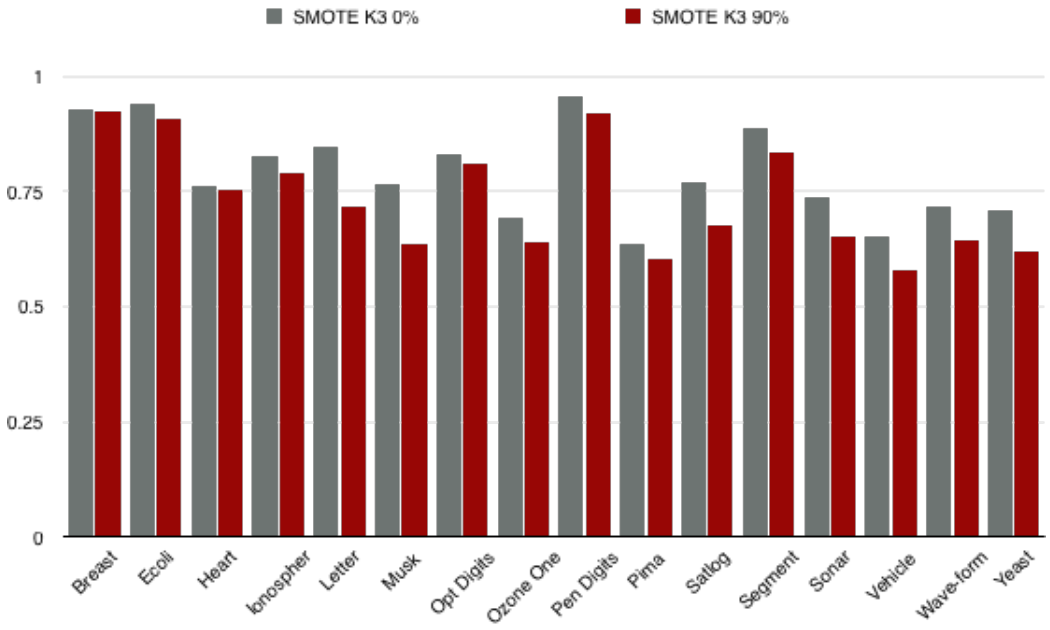


Figure B.5: Degradation of the mean of the classifiers with SMOTE k=3 after manifold augmentation on the UCI domains for $p = 0$ and $p = 90$.

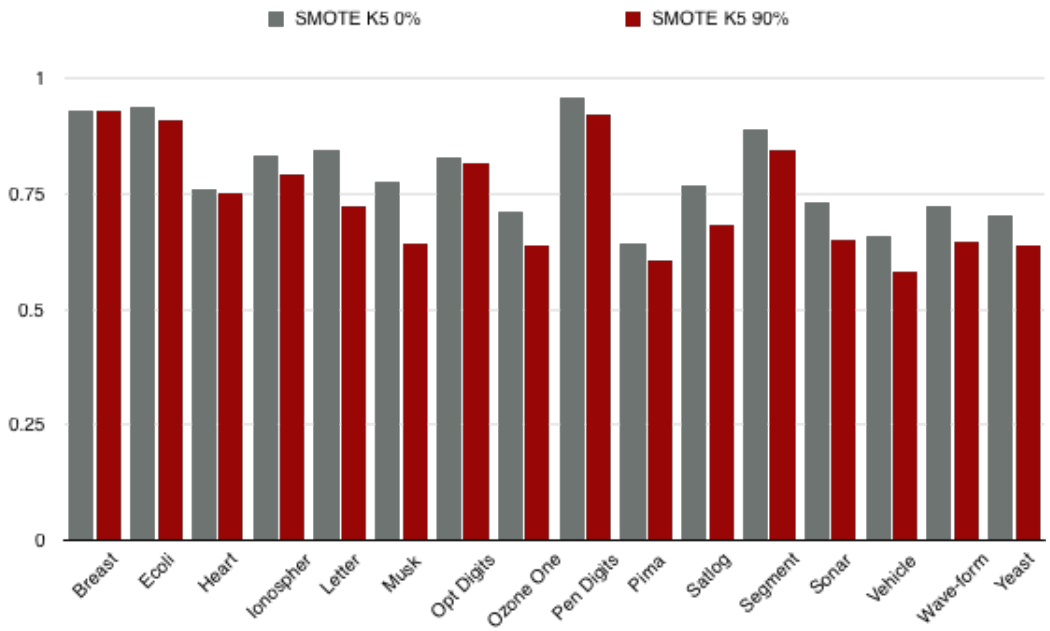


Figure B.6: Degradation of the mean of the classifiers with SMOTE k=5 after manifold augmentation on the UCI domains for $p = 0$ and $p = 90$.

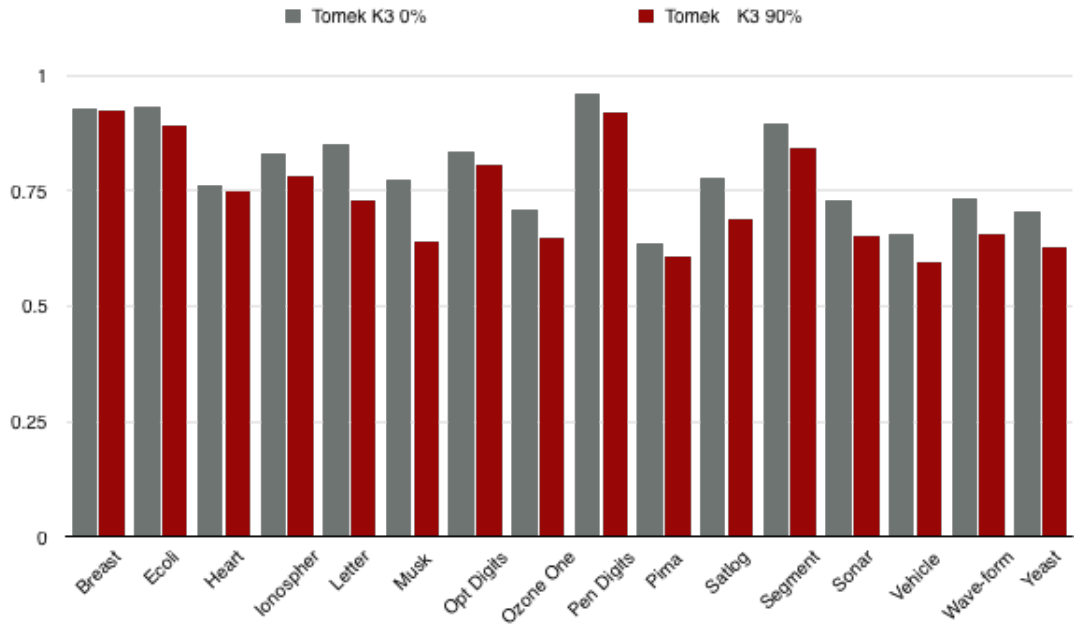


Figure B.7: Degradation of the mean of the classifiers with SMOTE+Tomek link k=3 after manifold augmentation on the UCI domains for $p = 0$ and $p = 90$.

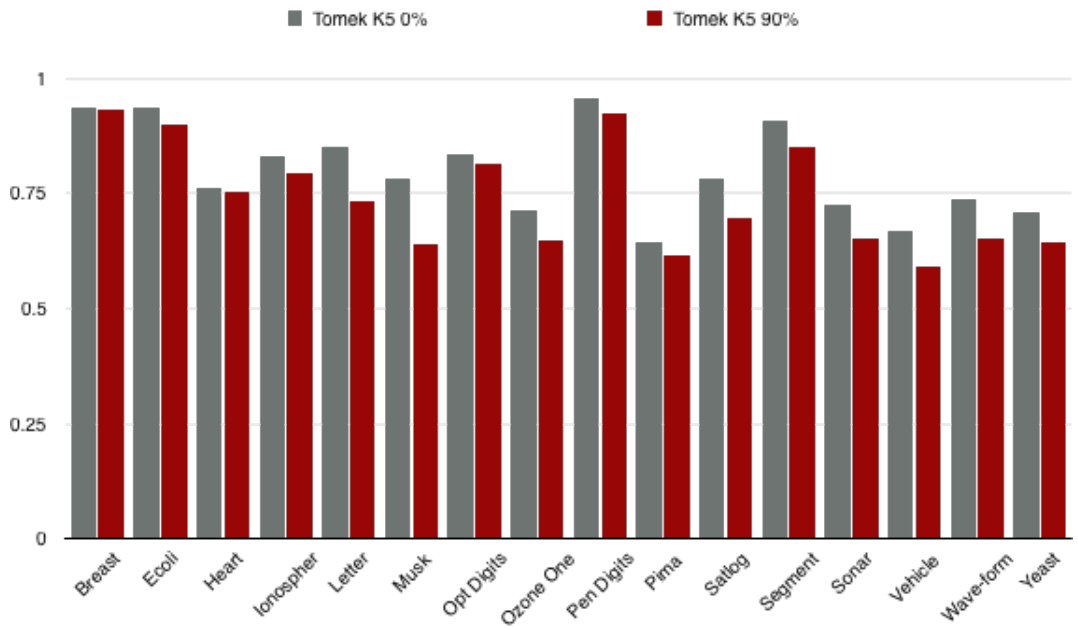


Figure B.8: Degradation of the mean of the classifiers with SMOTE+Tomek link k=5 after manifold augmentation on the UCI domains for $p = 0$ and $p = 90$.

Bibliography

[Els, 2014] (2014). Artificial Intelligence in Medicine.

[Eco, 2016] (2016). Digital Transformation.

[Aggarwal et al., 2001] Aggarwal, C. C., Hinneburg, A., Keim, D. a., Dethloff, G. E., Shin-Yi Lu, S., Hinneburg, A., Kol, I., Michor, P. W., Slov, J., Wien, A., Kushilevitz, E., Ostrovsky, R., Rabani, Y., Lee, D., Ondich, J., Rabitz, H., Alis, Ö., Al, Ö. F., Tresse, Wang, X., Li, Z., Zhang, L., Yuan, J., Acad, I., Melnick, K., Teichmann, J., Luzzatto, S., Tureli, S., War, K., Paul, S., Olver, P. J., Logemann, H., Ryan, E. P., Hermann, R., and Scott Krig (2001). On the Surprising Behavior of Distance Metrics in High Dimensional Space. *European Journal of Applied Mathematics*, 6(1):631–637.

[Akbani et al., 2004] Akbani, R., Kwek, S., and Japkowicz, N. (2004). Applying Support Vector Machines to Imbalanced Datasets. *Machine Learning: ECML 2004*, 3201(July):39–50.

[Alain and Bengio, 2014] Alain, G. and Bengio, Y. (2014). What Regularized Auto-Encoders Learn from the Data Generating Distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593.

[Allen, 2015] Allen, K. (2015). How a Toronto professor's research revolutionized artificial intelligence.

[Alpaydin, 2014] Alpaydin, E. (2014). *Introduction to Machine Learning*. Cambridge, Mass. : MIT Press, third edit edition.

[Anderson et al., 1977] Anderson, J. A., Silverstein, J. W., Ritz, S. A., and Jones, R. S. (1977). Distinctive features, categorical perception and probability learning: Some applications of a neural model. *Psychological Review*, 84:413–451.

[Baldi and Hornik, 1989] Baldi, P. and Hornik, K. (1989). Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53—-58.

[Batista et al., 2004a] Batista, G., Prati, R. C., and Monard, M. C. (2004a). A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd Explorations Newsletter*, 6(1):20–26.

[Batista et al., 2003] Batista, G. E. A. P. A., Bazzan, A. L. C., and Monard, M. C. (2003). Balancing Training Data for Automated Annotation of Keywords : a Case Study. In *Brazilian Workshop on Bioinformatics*, pages 10–18.

[Batista et al., 2004b] Batista, G. E. a. P. a., Prati, R. C., and Monard, M. C. (2004b). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKD Explorations Newsletter - Special issue on learning from imbalanced datasets*, 6(1):20.

[Batista et al., 2004c] Batista, G. E. a. P. a., Prati, R. C., and Monard, M. C. (2004c). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1):20.

[Belkin and Niyogi, 2003] Belkin, M. and Niyogi, P. (2003). Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15(6):1373–1396.

[Belkin and Niyogi, 2004] Belkin, M. and Niyogi, P. (2004). Semi-Supervised Learning on Riemannian Manifolds. *Machine Learning*, 56:209–239.

[Bellinger and Japkowicz,] Bellinger, C. and Japkowicz, N. Motivating the Inclusion of Meteorological Indicators in the CTBT Feature-Space. In *Proceedings of IEEE Symposium on Computational Intelligence for Security and Defense Applications*.

[Bellman, 1961] Bellman, R. (1961). *Adaptive Control Processes*. Princeton University Press.

[Blagus and Lusa, 2012] Blagus, R. and Lusa, L. (2012). Evaluation of SMOTE for High-Dimensional Class-Imbalanced Microarray Data. In *2012 11th International Conference on Machine Learning and Applications*, number 1, pages 89–94. Ieee.

[Breiman et al., 1984] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*.

[Buyting, 2015] Buyting, S. T. C. (2015). Deep Learning Godfather says machines learn like toddlers.

[Cattell, 1966a] Cattell, R. B. (1966a). The scree test for the number of factors. *Multivariate behavioral research*, 1(2).

[Cattell, 1966b] Cattell, R. B. (1966b). The scree test for the number of factors. *Multivariate behavioral research*, 1(2).

[Chapelle et al., 2006] Chapelle, O., Schölkopf, B., and Zien, A. (2006). *Semi-supervised Learning*. MIT Press.

[Chawla et al., 2002] Chawla, N., Bowyer, K., Hall, L., and W.P., K. (2002). SMOTE: Synthetic Minority Over-Sampling Technique. *J. Artificial Intelligence Research*, 16:321–357.

[Chawla et al., 2004] Chawla, N. V., Japkowicz, N., and Drive, P. (2004). Editorial : Special Issue on Learning from Imbalanced Data Sets. *ACM SIGKD Explorations Newsletter - Special issue on learning from imbalanced datasets*, 6(1):2000–2004.

[Chawla et al., 2003] Chawla, N. V., Lazarevic, A., Hall, L. O., and Bowyer, K. W. (2003). SMOTEBoost : Improving Prediction of the Minority Class in Boosting. In *Knowledge Discovery in Databases: PKDD*, pages 107–119.

[Clarkson, 1994] Clarkson, K. L. (1994). An algorithm for approximate closest-point queries. In *Proceedings of the tenth annual symposium on Computational geometry - SCG '94*, pages 160–164.

[Cohen et al., 2006] Cohen, G., Hilario, M., Sax, H., Hugonnet, S., and Geissbuhler, A. (2006). Learning from imbalanced data in surveillance of nosocomial infection. *Artificial Intelligence in Medicine*, 37(1):7–18.

[Courtney, 2013] Courtney, M. G. R. (2013). Determining the number of factors to retain in EFA : Using the SPSS R-Menu v2 . 0 to make more judicious estimations. *Practical Assessment, Research & Evaluation*, 18(8):1–14.

[Crawford and Ghosh, 2005] Crawford, M. M. and Ghosh, J. (2005). Applying nonlinear manifold learning to hyperspectral data for land cover classification. *Proceedings. 2005 IEEE International Geoscience and Remote Sensing Symposium, 2005. IGARSS '05.*, 6:4311–4314.

[Dietterich, 1998] Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923.

[Dillow, 2013] Dillow, C. (2013). The big data employment boom. *Fortune*.

[Drummond and Holte, 2000] Drummond, C. and Holte, R. C. (2000). Exploiting the cost (in) sensitivity of decision tree splitting criteria. In *International Conference on Machine Learning*, pages 239–246.

[Duda et al., 2001] Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. New York, Wiley, 2nd ed. edition.

[Elkan, 2001] Elkan, C. (2001). The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, pages 973–978.

[Fecker et al., 2013] Fecker, D., Märgner, V., and Fingscheidt, T. (2013). Density-induced oversampling for highly imbalanced datasets. In Bingham, P. R. and Lam, E. Y., editors, *SPIE. 8661, Image Processing: Machine Vision Applications VI 86610P*, volume 8661, pages 86610P–86610P–11.

[Feuersänger and Griebel, 2009] Feuersänger, C. and Griebel, M. (2009). Principal manifold learning by sparse grids. *Computing (Vienna/New York)*, 85(April):267–299.

[Gao et al., 2012] Gao, M., Hong, X., Chen, S., and Harris, C. J. (2012). Probability density function estimation based over-sampling for imbalanced two-class problems. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. Ieee.

[Garrido et al., 2011] Garrido, L. E., Abad, F. J., and Ponsoda, V. (2011). Performance of Velicer's minimum average partial factor retention method with categorical variables. *Educational and Psychological Measurement*.

[Goldberg et al., 2009] Goldberg, A. B., Zhu, X., Singh, A., Xu, Z., and Nowak, R. (2009). Multi-manifold semi-supervised learning. *Journal of Machine Learning Research*, 5:169–176.

[Guo et al., 2008a] Guo, G., Fu, Y., Dyer, C., and Huang, T. (2008a). Image-Based Human Age Estimation by Manifold Learning and Locally Adjusted Robust Regression. *IEEE Transactions on Image Processing*, 17(7):1178–1188.

[Guo et al., 2008b] Guo, X., Yin, Y., Dong, C., Yang, G., and Zhou, G. (2008b). On the Class Imbalance Problem. *2008 Fourth International Conference on Natural Computation*, pages 192–201.

[Ha and Bunke, 1997] Ha, T. M. and Bunke, H. (1997). Off-line, Handwritten Numeral Recognition by Perturbation Method. *Patter Analysis and Machine Intelligence*, 19(5):535–539.

[Hall et al., 2009] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

[Han et al., 2005] Han, H., Wang, W.-y., and Mao, B.-h. (2005). Borderline-SMOTE : A New Over-Sampling Method in Imbalanced Data Sets Learning. *Advances in intelligent computing*, pages 878–887.

[Hand and Vinciotti, 2003] Hand, D. J. and Vinciotti, V. (2003). Local Versus Global Models for Classification Problems. *The American Statistician*, 57(2):124–131.

[Hart, 1968] Hart, P. E. (1968). The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, IT-14:515–516.

[He et al., 2008] He, H., Bai, Y., Garcia, E. A., and Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, (3):1322–1328.

[He and Garcia, 2009] He, H. and Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.

[Hebb, 1949] Hebb, D. (1949). *The Organization of Behavior*. New York, Wiley.

[Hernandez, 2014] Hernandez, D. (2014). Meet the Man Google Hired to Make AI a Reality. *Wired Magazine*.

[Hinton and Salakhutdinov, 2006] Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the Dimensionality of. *Science (New York, N.Y.)*, 313(July):504–507.

[Hof, 2014] Hof, R. (2014). Interview: Inside Google Brain Founder Andrew Ng's Plans To Transform Baidu.

[Holte et al., 1989] Holte, R. C., Acker, L. E., and Porter, B. W. (1989). Concept Learning and the Problem of Small Disjuncts. *IJCAI*, 89:813–818.

[Horn, 1965] Horn, J. L. (1965). A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30:179–185.

[Hughes, 1968] Hughes, G. (1968). On the mean accuracy of statistical pattern recognizers. *Information Theory, IEEE Transactions on*, 14(1):55–63.

[Huo et al., 2007] Huo, X., Ni, X. S., and Smith, A. K. (2007). A Survey of Manifold-Based Learning Methods. In *Recent advances in data mining of enterprise data*, pages 691–745.

[Izenman, 2011] Izenman, A. J. (2011). Spectral Embedding Methods for Manifold Learning. In *Manifold Learning Theory and Applications*, pages 1–36.

[Izenman, 2012] Izenman, A. J. (2012). Introduction to manifold learning. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(5):439–446.

[Japkowicz, 2000] Japkowicz, N. (2000). earning from Imbalanced Data Sets. In *the American Association for Artificial Intelligence*. (Technical Report WS-00-05).

[Japkowicz, 2001] Japkowicz, N. (2001). Supervised versus unsupervised binary-learning by feedforward neural networks. *Machine Learning*, 42(1):97–122.

[Japkowicz and Stephen, 2002] Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449.

[Jia Wei et al., 2008] Jia Wei, Hong Peng, Yi-Shen Lin, Zhi-Mao Huang, and Jia-Bing Wang (2008). Adaptive neighborhood selection for manifold learning. In *2008 International Conference on Machine Learning and Cybernetics*, volume 1, pages 380–384.

[Jo and Japkowicz, 2004] Jo, T. and Japkowicz, N. (2004). Class Imbalances versus Small Disjuncts. *ACM SIGKDD Explorations Newsletter - Special issue on learning from imbalanced datasets*, 6(1):40–49.

[Juanjuan et al., 2007] Juanjuan, W., Mantao, X., Hui, W., and Jiwu, Z. (2007). Classification of imbalanced data by using the SMOTE algorithm and locally linear embedding. In *International Conference on Signal Processing Proceedings, ICSP*, volume 3, pages 1–4.

[Kaiser, 1960] Kaiser, H. F. (1960). The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, 20:141–151.

[Kangas et al., 2008] Kangas, L. J., Keller, P. E., Siciliano, E. R., Kouzes, R. T., and Ely, J. H. (2008). The use of artificial neural networks in PVT-based radiation portal monitors. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 587(2-3):398–412.

[Kohonen, 1977] Kohonen, T. (1977). *Associative memories*. Berlin: Springer.

[Kosner, 2014] Kosner, A. W. (2014). Tech 2015: Deep Learning And Machine Intelligence Will Eat The World. *Forbes Magazine*.

[Kubat et al., 1998] Kubat, M., Holte, R. C., and Matwin, S. (1998). Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Machine Learning*, 30(2-3):195–215.

[Kubat and Matwin, 1997] Kubat, M. and Matwin, S. (1997). Addressing the curse of imbalanced data sets: One-sided sampling. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 179–186.

[Laurikkala, 2001] Laurikkala, J. (2001). Improving Identification of Difficult Small Classes by Balancing Class Distribution. In *AI in Medicine in Europe: Artificial Intelligence Medicine*, pages 63–66. University of Tampere.

[LeCun et al., 2006] LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. (2006). A tutorial on energy-based learning. *Predicting structured data*, 1:1–59.

[Lewis et al., 1994] Lewis, D. D., Catlett, J., and Hill, M. (1994). Heterogeneous Uncertainty Sampling for Supervised Learning. In *International Conference on Machine Learning*, pages 148–156.

[Li, 2006] Li, L. (2006). Data complexity in machine learning and novel classification algorithms. Technical Report May, Caltech.

[Liu et al., 2007] Liu, A., Ghosh, J., and Martin, C. E. (2007). Generative Oversampling for Mining Imbalanced Datasets. In *International Conference on Data Mining*, pages 66–72.

[Ma and Fu, 2011] Ma, Y. and Fu, Y. (2011). *Manifold Learning Theory and Applications*.

[Maloof et al., 1997] Maloof, M., Langley, P., Sage, S., and Binford, T. (1997). Learning to Detect Rooftops in Aerial Images. In *Proc. Image Understanding Workshop*, pages 835–845.

[Maloof, 2003] Maloof, M. A. (2003). Learning when data sets are imbalanced and when costs are unequal and unknown. In *ICML-2003 workshop on learning from imbalanced data sets II*, pages 1–2.

[McCallum and Nigam, 1998a] McCallum, A. and Nigam, K. (1998a). A Comparison of Event Models for Naive Bayes Text Classification. In *AAAI Workshop on Learning for Text Categorization*, pages 41–48.

[McCallum and Nigam, 1998b] McCallum, A. and Nigam, K. (1998b). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, pages 41–48.

[McCarthy et al., 2005] McCarthy, K., Zabar, B., and Weiss, G. (2005). Does cost-sensitive learning beat sampling for classifying rare classes? In *Proceedings of the 1st international workshop on Utility-based data mining*, pages 69–77.

[McClelland and Rumelhart, 1986] McClelland, J. L. and Rumelhart, D. E. (1986). A distributed model of memory. *Parallel distributed processing: Explorations in the microstructure of cognition*, 2(Applications):170–215.

[McCulloch and Pitts, 1943] McCulloch, W. and Pitts, W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 5(4):115–133.

[Melville and Mooney, 2004] Melville, P. and Mooney, R. J. (2004). Diverse ensembles for active learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*.

[Mika et al., 1999] Mika, S., Schölkopf, B., Smola, A., Müller, K.-R., Scholz, M., and Rätsch, G. (1999). Kernel PCA and De-Noising in Feature Spaces. *Analysis*, 11(i):536–542.

[Mitchell, 1980] Mitchell, T. (1980). The Need for Biases in Learning Generalizations. Technical report, Rutgers Computer Science Department Technical Report CBM-TR-117.

[Mitchell, 1997] Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.

[Olmos et al., 1991] Olmos, P., Diaz, J., Perez, J., Gomez, P., Rodellar, V., Aguayo, P., Bru, A., Garcia-Belmonte, G., and de Pablos, J. (1991). A new approach to automatic radiation spectrum analysis. *Nuclear Science, IEEE Transactions on*, 38(4):971–975.

[Olshausen and Fieldt, 1997] Olshausen, B. A. and Fieldt, D. J. (1997). Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1. *Vision Research*, 37(23):3311–3325.

[Pan et al., 2009] Pan, Y., Ge, S. S., and Al Mamun, A. (2009). Weighted locally linear embedding for dimension reduction. *Pattern Recognition*, 42(5):798–811.

[Pearson, 1901] Pearson, K. (1901). On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine*, 2(11):559–572.

[Provost and Fawcett, 2001] Provost, F. and Fawcett, T. (2001). Robust Classification for Imprecise Environments. *Machine Learning*, 42(3):203–231.

[Provost and Fawcett, 1997] Provost, F. J. and Fawcett, T. (1997). Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In *AAAI*, pages 43–48.

[Ranzato et al., 2008] Ranzato, M., Boureau, Y.-L., and LeCun, Y. (2008). Sparse feature learning for deep belief networks. In *NIPS'07*, pages 1185–1192.

[Rifai et al., 2012] Rifai, S., Bengio, Y., Dauphin, Y., and Vincent, P. (2012). A Generative Process for Sampling Contractive Auto-Encoders. In *International Conference on Machine Learning*, number 1. arXiv preprint arXiv:1206.6434.

[Roweis and Saul, 2000] Roweis, S. and Saul, L. (2000). Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326.

[Rumelhart et al., 1995] Rumelhart, D. E., Durbin, R., Golden, R., and Chauvin, Y. (1995). Backpropagation: The basic theory. In *Mathematical perspectives on neural networks*, pages 533–566.

[Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. *Parallel distributed processing: Explorations in the microstructure of cognition1*, 1:318–362.

[Ruscio and Roche, 2012] Ruscio, J. and Roche, B. (2012). Determining the number of factors to retain in an exploratory factor analysis using comparison data of known factorial structure. *Psychological assessment*, 24(2):282.

[Saul and Roweis, 2003] Saul, L. K. L. and Roweis, S. S. T. (2003). Think globally, fit locally: unsupervised learning of low dimensional manifolds. *The Journal of Machine Learning Research*, 4:119–155.

[Schölkopf and Burges, 1999] Schölkopf, B. and Burges, J. (1999). *Advances in kernel methods: support vector learning*. MIT Press.

[Sen et al., 2008] Sen, S. K., Foskey, M., Marron, J. S., and Styner, M. (2008). Support vector machine for data on manifolds: An application to image analysis. In *In Biomedical Imaging: From Nano to Macro*, pages 1195–1198.

[Shao et al., 2014] Shao, K., Zhai, Y., Sui, H., Zhang, C., and Ma, N. (2014). A New Over-sample Method Based on Distribution Density. *Journal of Computers*, 9(2):483–490.

[Sharma et al., 2012a] Sharma, S., Bellinger, C., and Japkowicz, N. (2012a). Clustering based one-class classification for the comprehensive nuclear test-ban treaty. In Kosseim, Leila and Inkpen, D., editor, *Advances in Artificial Intelligence, Lecture Notes in Computer Science*, volume 7310, pages 181–193. Springer Berlin / Heidelberg.

[Sharma et al., 2012b] Sharma, S., Bellinger, C., Japkowicz, N., Berg, R., and Ungar, K. (2012b). Anomaly detection in gamma ray spectra: A machine learning perspective. *2012 IEEE Symposium on Computational Intelligence for Security and Defence Applications*, 1(i):1–8.

[Silva and Tenenbaum, 2002] Silva, V. D. and Tenenbaum, J. B. (2002). Global versus local methods in nonlinear dimensionality reduction. *Advances in neural information processing systems*, pages 705–712.

[Simonite, 2014] Simonite, T. (2014). Chinese Search Giant Baidu Hires Man Behind the Google Brain. *MIT Technology Review*.

[Sparks and Madabhushi, 2013] Sparks, R. and Madabhushi, A. (2013). Statistical shape model for manifold regularization: Gleason grading of prostate histology. *Computer Vision and Image Understanding*, 117(9):1138–1146.

[Srivastava et al., 2014] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

[Stefanowski and Wilk, 2007] Stefanowski, J. and Wilk, S. (2007). Improving rule-based classifiers induced by MODLEM by selective pre- processing of imbalanced data. In *ECML/PKDD international workshop on rough sets in knowledge discovery (RSKD'2007)*, pages 54–65.

[Stocki et al., 2010] Stocki, T. J., Li, G., Japkowicz, N., and Ungar, R. K. (2010). Machine learning for radioxenon event classification for the Comprehensive Nuclear-Test-Ban Treaty. *Journal of environmental radioactivity*, 101(1):68–74.

[Sun et al., 2007] Sun, Y., Kamel, M., Wong, A., and Wang, Y. (2007). Cost-Sensitive Boosting for Classification of Imbalanced Data. *Pattern Recogni- tion*, 40(12):3358–3378.

[Tomek, 1976] Tomek, I. (1976). Modifications of CNN. *IEEE Trans. System, Man, Cybernetics*, 6(11):769–772.

[Tuzel et al., 2007] Tuzel, O., Porikli, F., and Mee, P. (2007). Human detection via classification on Riemannian manifolds. In *Computer Vision and Pattern Recognition*, pages 1–8.

[van den Bosch et al., 1997] van den Bosch, A., Weijters, T., van den Herik, H. J., and Daelemans, W. (1997). When small disjuncts abound, try lazy learning: A case study. In *the Seventh Belgian- Dutch Conference on Machine Learning*, pages 109–118.

[van der Maaten et al., 2009] van der Maaten, L., Postma, E., and van den Herik, J. (2009). Dimensionality Reduction: A Comparative Review. *Advances in Neural Information Processing*, (February).

[Vapnik, 1995] Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag., New York.

[Vincent et al., 2008] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P. A. (2008). Extracting and composing robust features with denoising autoencoders. In *25th international conference on Machine learning*, pages 1096–1103.

[Vincent et al., 2010] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked Denoising Autoencoders : Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *The Journal of Machine Learning Research*, 11:3371–3408.

[Wallace et al., 2011] Wallace, B. C., Small, K., Brodley, C. E., and Trikalinos, T. a. (2011). Class Imbalance, Redux. In *2011 IEEE 11th International Conference on Data Mining*, pages 754–763. Ieee.

[Wang and Japkowicz, 2010] Wang, B. X. and Japkowicz, N. (2010). Boosting support vector machines for imbalanced data sets. *Knowledge and Information Systems*, 25(1):1–20.

[Weinberger et al., 2004] Weinberger, K. Q., Sha, F., and Saul, L. K. (2004). Learning a kernel matrix for nonlinear dimensionality reduction. In *International Conference on Machine Learning*, pages 106–113.

[Weinberger and Saul, 2004] Weinberger, Q. and Saul, L. (2004). Unsupervised Learning of Image Manifolds by Semidefinite Programming. *Cvpr*, 70(1):988–995.

[Weiss, 2004] Weiss, G. M. (2004). Mining with Rarity : A Unifying Framework. *SIGKDD Explor. Newsl.*, 6(1):7–19.

[Williams et al., 2009] Williams, D., Myers, V., and Silvious, M. (2009). Mine Classification With Imbalanced Data. *IEEE Geoscience and Remote Sensing Letters*, 6(3):528–532.

[Wilson, 1972] Wilson, D. (1972). Asymptotic Properties of Nearest Neighbor Rules using Edited Data. *IEEE Trans. on Systems, Man and Cybernetics*, 2:408–420.

[Wolpert, 1996] Wolpert, D. (1996). The Lack of A Priori Distinctions between Learning Algorithms. *Neural Computation*, pages 1341–1390.

[Wong, 2014] Wong, G. (2014). Baidu's Andrew Ng on Deep Learning and Innovation in Silicon Valley. *The Wall Street Journal*.

[Xu et al., 2010] Xu, Z., King, I., Lyu, M. R. T., and Jin, R. (2010). Discriminative semi-supervised feature selection via manifold regularization. *IEEE Transactions on Neural Networks*, 21(7):1033–1047.

[Xue and Chen, 2007] Xue, H. U. I. and Chen, S.-c. (2007). Alternative robust local embedding. In *2007 International Conference on Wavelet Analysis and Pattern Recognition*, pages 591–596.

[Yang et al., 2006] Yang, Q., Wu, X., Elkan, C., Gehrke, J., Han, J., Heckerman, D., Keim, D., Liu, J., Madigan, D., Piatetsky-shapiro, G., Raghavan, V. V., Rastogi, R., Stolfo, S. J., Tuzhilin, A., and Wah, B. W. (2006). 10 challenging problems in data mining research. 5(4):597–604.

[Yoshida et al., 2002] Yoshida, E., Shizuma, K., Endo, S., and Oka, T. (2002). Application of neural networks for the analysis of gamma-ray spectra measured with a Ge spectrometer. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 484(1-3):557–563.

[Yu et al., 2009] Yu, K., Zhang, T., and Gong, Y. (2009). Nonlinear Learning using Local Coordinate Coding. In *Nips*, pages 2223–2231.

[Zhang and Chen, 2005] Zhang, D. and Chen, X. (2005). Text Classification with Kernels on the Multinomial Manifold. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 266–273.

[Zhang and Wang, 2011] Zhang, L. and Wang, W. (2011). A Re-sampling Method for Class Imbalance Learning with Credit Data. In *2011 International Conference of Information Technology, Computer Engineering and Management Sciences*, pages 393–397. Ieee.

[Zhang and Zha, 2004] Zhang, Z. and Zha, H. (2004). Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignment. *Journal of Shanghai University (English Edition)*, 8(4):406–424.