Mach Learn (2018) 107:605–637
https://doi.org/10.1007/s10994-017-5670-4

Manifold-based synthetic oversampling with manifold conformance estimation

Colin Bellinger[1] · **Christopher Drummond**[3] ·
Nathalie Japkowicz[2]

Received: 12 September 2016 / Accepted: 6 September 2017 / Published online: 19 October 2017

Abstract Classification domains such as those in medicine, national security and the environment regularly suffer from a lack of training instances for the class of interest. In many cases, classification models induced under these conditions have poor predictive performance on the important minority class. Synthetic oversampling can be applied to mitigate the impact of imbalance by generating additional training instances. In this field, the majority of research has focused on refining the SMOTE algorithm. We note, however, that the generative bias of SMOTE is not appropriate for the large class of learning problems that conform to the manifold property. These are high-dimensional problems, such as image and spectral classification, with implicit feature spaces that are lower-dimensional than their physical data spaces. We show that ignoring this can lead to instances being generated in erroneous regions of the data space. We propose a general framework for manifold-based synthetic oversampling that helps users to select a domain-appropriate manifold learning method, such as PCA or autoencoder, and apply it to model and generate additional training samples. We evaluate data generation on theoretical distributions and image classification tasks that are standard in the manifold learning literature, and empirically show its positive impact on the classification of high-dimensional image and gamma-ray spectra tasks, along with 16 UCI datasets.

Editor: Johannes Fürnkranz.

✉ Colin Bellinger
cbelling@ualberta.ca

Christopher Drummond
Christopher.Drummond@nrc-cnrc.gc.ca

Nathalie Japkowicz
nathalie.japkowicz@american.edu

[1] Department of Computing Science, University of Alberta, Edmonton, Canada

[2] Department of Computer Science, American University, Washington, DC, USA

[3] National Research Council of Canada, Ottawa, Canada

Keywords Class imbalance · Synthetic oversampling · Manifold learning · SMOTE

1 Introduction

Problems such as radioactive threat classification, oil spill classification, gene function annotation, and medical and text classification have challenging properties (Bellinger et al. 2015; Kubat et al. 1998; Blondel et al. 2011; Akbani et al. 2004; Nguwi and Cho 2009). These domains have a high level of complexity due to factors such as class overlap and multimodality. Moreover, the class distributions are imbalanced and the minority classes are rare. The degree of rarity can be thought of as a function of the number of training examples relative to the complexity of the domain.

Rarity creates a challenging learning scenario referred to as absolute imbalance. Learners trained on absolutely imbalanced data are known to produce error prone predictions (He and Garcia 2009). For an arbitrary dataset, the specific threshold below which a domain is said to be absolutely imbalanced is unclear, but the outcome is not. It breaks the general assumption of machine learning that demands a representative set of instances from each class. An absolutely imbalanced training set leads to the induction of a decision boundary that is biased in favour of the majority class, thereby causing weak classification accuracy (Jo and Japkowicz 2004; Weiss 2004; He and Garcia 2009).

Given the practical importance, and the significant challenge posed by domains of this nature, class imbalance has been identified as one of the essential problems in machine learning (Yang et al. 2006) and has spawned workshops, conferences and special issues to discuss and develop strategies to manage class imbalance (Japkowicz 2000; Chawla et al. 2003, 2004; Chawla and Zhou 2009; Wang et al. 2017).

The most obvious solution to the problem of class imbalance is more training samples. Unfortunately, we cannot sample the data directly due to domain properties, such as acquisition cost and class probability. Thus, we turn to the data-driven generation of synthetic instances. Within class imbalance, this is known as synthetic oversampling, and was originally achieved by interpolating new instances at random distances between nearest neighbours in the minority class; this is the classic SMOTE algorithm (Chawla et al. 2002).

SMOTE has been very successful in reducing the impact of class imbalance, and as such, it has seen wide-spread application. Nonetheless, many studies have also found that it does not always generate synthetic instances that produce the desired performance gains (Drummond and Holte 2003; Van Hulse et al. 2007; Wallace et al. 2011). In turn, these studies have lead to the development of new variations on the standard algorithm, such as boosting, post-hoc cleaning and better methods for selecting the instances to be used for generation by SMOTE (Chawla et al. 2003; Han et al. 2005; Bunkhumpornpat et al. 2009).

In spite of the various proposed alterations, the SMOTE algorithm and its derivatives suffer from a particular and identifiable weakness on a great number of domains that are of particular importance in modern machine learning, such as those involving text, speech, video, image, radiation spectra and RNA. Each of these domains, and many more like them, are of high-dimensionality. Methods, such as the SMOTE algorithm, that rely on calculations of the straight-line distance between points in high-dimensional spaces are known to be error prone. These error prone distances lead to inaccurate nearest neighbour selection, and thus, the generation of noisy synthetic instances. These negatively impact the performance of the induced classifier. Removing and/or avoiding these noisy instances is the focus of the previously mentioned derivatives of the SMOTE algorithm. The outstanding problem, however, is that it is difficult to determine which approach is ideal, and more importantly, none of these directly deals with the core of the problem.

We argue that a superior means of synthesizing instances for high-dimensional class imbalance tasks is to take advantage of the intrinsic structure of the data. In particular, we note that the class probability mass is not spread widely throughout the high-dimensional space, but rather embedded in a much lower-dimensional manifold-space (Chapelle et al. 2006). Harnessing the intrinsic manifold property has been key to improving performance in many high-value machine learning domains, such as image, speech, and text classification, human action and emotion recognition in audio and video, and is necessary to advance performance on related imbalanced domains (Tuzel et al. 2007b, 2008; Lui et al. 2010; Liu et al. 2013; Slama et al. 2015).

The explicit weakness of SMOTE, and its related algorithms, is that it operates in the fog of the higher-dimensional data-space. The curvature of the embedded manifold causes orthogonal instances to appear deceptively close whilst their geodesic distance along the manifold is, in fact, great. The direct consequence of this is that non-manifold-based synthetic oversampling of domains that conform to the manifold property will inherently lead to noisy instances that negatively impact performance.

On the surface, feature selection can seem like a reasonable alternative to manifold learning. Feature selection methods, such as subset selection, choose a subset of the d-dimensions to represent the data (Alpaydin 2014). This is not an effective means of solving problems in domains that conform to the manifold property because the probably density resides in an embedded space. This cannot be recovered by extracting a subset of features from the feature space. Moreover, many selected features methods themselves can be biased by the skewed distribution.

Indeed, through our practical experience in applying synthetic oversampling methods to gamma-ray spectral classification problems, we were able to identify that SMOTE and other synthetic oversampling methods perform poorly when the target data conforms to the manifold property. In particular, they have the tendency to synthesize noisy instances that are realized as distorted images and unrealistic gamma-ray spectra.

To address this, we propose a framework for synthetically oversampling data that conforms to the manifold property. The schema for the four component framework is presented in Fig. 1. In brief, the framework evaluates the conformance of the domain to the manifold property to determine an appropriate means of synthetic oversampling. Based on this outcome, an appropriate manifold learning method can be selected and applied to the data. The synthetic instances are then generated in the manifold-space and reverse mapped back to the feature space to be added to the training set.

The contributions of this work are to: (a) demonstrate the weakness of existing methods when the data conforms to the manifold property, (b) develop a framework for manifold-based synthetic oversampling that can utilize any manifold learning algorithm, (c) propose and evaluate methods for quantifying conformance to the manifold property to determine when manifold-based synthetic oversampling is ideal, (d) demonstrate two formalizations of the framework (autoencoder and PCA) and suggest others, and (e) empirically show the benefits of the framework on high-dimensional, real-world data that conforms to the manifold property. In particular, our experiments include high-dimensional image classification and gamma-ray spectra classification tasks, along with 16 benchmark UCI datasets. These contributions bring together and further elaborate on results that we have previously reported (Bellinger et al. 2015, 2016).

In this work, our proposed method for testing the conformance of datasets to the manifold property is added to improve the usability of the framework. It determines if manifold-based

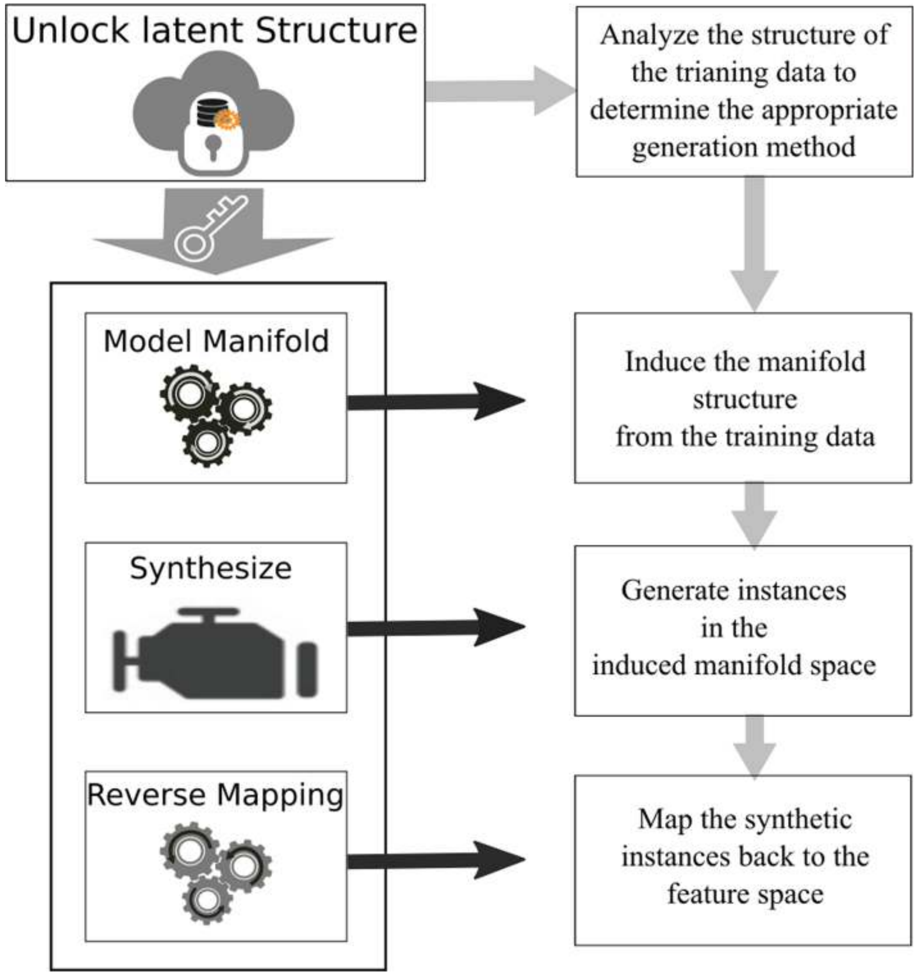

Unlock latent Structure
Analyze the structure of the trianing data to determine the appropriate generation method
Model Manifold
Induce the manifold structure from the training data
Synthesize
Generate instances in the induced manifold space
Reverse Mapping
Map the synthetic instances back to the feature space

Fig. 1 General framework for synthetic oversampling

synthetic oversampling is ideal for the target domain. Moreover, it offers the potential to aid in the choice of which manifold learning method to utilize. These are two fundamental questions that were left unanswered in our previous work.

2 Related work

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, it is known to suffer from high variance and has the potential to discard valuable information from the majority class, whilst still risking overfitting the minority class (Han et al. 2005). Bagged random undersampling was proposed to deal with the high degree of variance (Wallace et al. 2011). This does not, however, address the significant loss of information. In the case of absolute imbalance, for example, the amount of undersampling needed would risk the majority class also becoming rare.

Alternatively, cost-sensitive learning does not necessitate the discarding of information. It aims to assign misclassification costs in a manner that promotes the induction of a classifier with improved accuracy on the minority class. The idea is that if you skew the misclassification costs towards the minority class, the classifier learns to make minority class predictions that it might not make otherwise. The set of cost-sensitive methods can be roughly separated into those that apply misclassification costs to the training data, those that utilize cost-sensitivity in conjunction with ensemble methods, and those that incorporate cost-sensitivity into the learning algorithms themselves (He and Garcia 2009). In practice, these methods can be challenging to apply due to their algorithmic forms and the need to estimate costs.

A more significant issue with cost-sensitive learning is that it relies on misclassifications during the inductive process to adjust the decision boundary. If no training instances are classified incorrectly, the cost adjustment has no impact on the induced classifier (Wallace et al. 2011). This scenario is not as improbable as it may seem. When the minority class is rare and/or the data is high-dimensional, it is possible that a decision boundary can easily be found that separates the two classes. Due to rarity, this decision boundary is unlikely to accurately represent the latent distribution, thus, leaving an inaccurate classifier in place.

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 and avoids discarding informative majority class instances; however, it does not prevent overfitting (Batista et al. 2004a). Synthetic oversampling was envisioned as a means of avoiding overfitting by generating synthetic instances to populate the minority class rather than by simply replicating instances (Chawla et al. 2002). Therefore, it does not necessitate the discarding of useful instances from the majority class, and when an appropriate bias is selected, it offers the potential to accurately adjust the decision boundary regardless of the separability of the training set.

The state-of-the-art methods in synthetic oversampling are primarily based on the SMOTE algorithm, which applies a generative bias that assumes the best place to generate synthetic instances is between nearest neighbours in the minority class. As a result, the synthetic set is generated entirely within the convex hull formed by these instances. There are two major criticisms of SMOTE. The first is that the instances are erroneously generated within the majority space. This causes the inductive process to overcompensate for the prediction bias of the classifier. The other, contradictory, criticism is that it does not generate instances close enough to the majority class. As a result, the prediction bias is not sufficiently reduced. The properties of the data domain determine which of these criticisms applies to SMOTE for a given domain.

A series of ad-hoc modifications to SMOTE have been proposed in an attempt to manage its weaknesses. Post-hoc methods have been described that attempt to remove minority instances generated in the majority space (Batista et al. 2004a; Han et al. 2005; Stefanowski and Wilk 2007). Other methods have been developed that aim to promote the generation of instances close to the majority space (Batista et al. 2003; He et al. 2008).

Nonetheless, the weaknesses of SMOTE and its derivatives relate to its generative bias, i.e., whether the convex hull formed via straight-line measurements in the feature space accurately covers the minority class. This is a reasonable assumption for many low- and medium-dimensional domains. As the dimensionality and corresponding sparsity increases, however, SMOTE becomes a limiting choice. To this end, our research shows that the existing means of synthetic oversampling are inappropriate for data that conforms to the manifold property, and that manifold-based synthetic oversampling should be applied.

3 Problem overview

Our research was originally inspired by our collaboration with the Radiation Protection Bureau at Health Canada, where we applied machine learning for safety in regards to radiation. The primary challenges were the high-dimensionality of the domain and the degree of imbalance. These are features that are common to a large number of classification domains, such as global climate change, image recognition, human identification, text classification and spectral classification.

We recognized that domains with this property can often be better represented in a lower-dimensional embedded space. This concept takes advantage of the reality that instances are not spread throughout the feature space but are concentrated around a lower-dimensional manifold. A simple example of a manifold in machine learning comes from handwritten digit recognition, where the digits are recorded in a high-dimensional feature space, but can be effectively represented in a lower-dimensional embedded space (Domingos 2012). Manifold learning provides a gateway to the embedded space in which all possible handwritten digits can be encoded.

Somewhat similar to the variations caused by an individuals style of writing on the shape and orientation of handwritten digit, the data distribution representing a particular isotope of interest, such as cobalt-60, is impacted by a combination of factors. When measured with a gamma-ray spectrometer, a pure isotope sample produces a very specific signature in the gamma-ray spectra. This clear signal, however, is altered and eroded by the amount of radioactive material, the degree to which it is shielded, the amount of the background radiation, and the distance between the source and the gamma-ray spectrometer. Different combinations of these aspects may affect a signal at any given time. The various combinations that may occur in the high-dimensional space can be imagined as forming a dense, low-dimensional structure in the embedded space.

Inducing a manifold representation of the training data provides a gateway to an embedded space that concisely represents the various forms of gamma-ray spectra that may occur in the class of interest. These include combinations of its different degraded and eroded states. If you can imagine taking a walk along the induced manifold, as you travel from one end to the other, you would be traveling along a continuum representing the transition between the different aspects that affect the spectra, including the degree of shielding, amount of radioactive material, distance to the source, etc.

A significant amount of research has been dedicated to the development of manifold learning methods (Huo et al. 2007; Ma and Fu 2011). The resulting algorithms utilize a diverse set of assumptions and biases, such as the complexity of the curvature of the manifold and the nature of the noise. Classic methods, such as PCA and MDS, are simple and efficient. These are guaranteed to determine the structure of the data on or near the embedded manifold. These traditional methods assume a linear manifold (Tenenbaum et al. 2000). Other, more algorithmically complex methods, such as kernel PCA and autoencoding, enable the induction of non-linear manifolds. Manifold learning has demonstrated great potential in clustering, classification and dimension reduction (Belkin and Niyogi 2003; Tuzel et al. 2007a; Zhang and Chen 2005; Roweis and Saul 2000). In spite of its potential, manifold learning methods have gone unconsidered in problems of class imbalance. We address this gap in the literature with a framework for manifold-based synthetic oversampling. This enables the user to select the most appropriate manifold learning strategy, be it a simple and efficient linear approach like PCA or local linear embedding, or more sophisticated non-linear methods such as the autoencoder.

3.1 Limitations of SMOTE on manifolds

Chawla et al. presented Synthetic Minority Oversampling TEchnique (SMOTE) (Chawla et al. 2002) as a general purpose synthesizer to overcome the limitation of random oversampling. SMOTE augments the minority training set by interpolating points between nearest neighbours in S_{min}. For each $x_i \in S_{min}$, a synthetic instance 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

$$x_{new} = x_i + (x_j - x_i) \times \delta. \tag{1}$$

To reduce the risk of synthesizing instances too deep inside the majority space, post-hoc processing to remove instances that form Tomek links has been proposed for SMOTE (Batista et al. 2003, 2004b). A pair (x_i, x_j) is considered to be a Tomek link 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 link, then either x_i or x_j are noise or they are on the class border, and as such they are deleted. This has the effect of shifting the convex hull away from the majority class. Alternatively, it is sometimes desired that the convex hull be shifted towards the majority class. Borderline SMOTE has been proposed to do this (Han et al. 2005). It finds a subset $S'_{min} \in S$ such that the elements of S'_{min} are on the border of the majority class. SMOTE is then applied to the instances in the subset S'_{min}.

The two major issues with SMOTE on manifold data are that (a) it is applied in the fog of the higher-dimensional feature-space, and (b) it applies a straight-line distance, typically calculated with the Minkowski distance, to find nearest neighbours. This is error prone for absolutely imbalanced data because the instances are expected to be far apart in the feature space. In Fig. 2, we illustrate the weaknesses of SMOTE by using a one-dimensional manifold embedded in a two-dimensional space. In the demonstration, we use the general version of SMOTE. This is warranted because the more recent adaptations apply the same bias, they suffer from the same weaknesses on data that conforms to the manifold property.

The top left graphic in Fig. 2 shows the manifold in red with samples from the manifold appearing as black circles. Each instance can be represented by its one-dimensional coordinate m in the manifold space. In machine learning, we often have data in the fog of a higher-dimensional feature space, not the embedded space. Manifold learning induces a model of the embedded space, and from this we can focus the generation of instances in high probability regions. This is visualized in the top right graphic, where the blue shading illustrates the probability mass being spread along the manifold. In the subsequent section, we demonstrate how this is achieved with our proposed framework.

The bottom graphics demonstrate the result of synthetic oversampling using SMOTE with $k = 7$ and $k = 3$. The first thing that we note is that the synthesized instances are clustered in many small pockets for the smaller k-value. This value results in the creation of a set of small, dense and disjunct convex hulls. These clusters fail to add much information specifically because they are tightly clustered around training instances that are already available. The larger k-value creates one large, uniformly populated convex hull. This results from the fact that for larger k-values the algorithm must search increasingly greater distances to find so-called nearest-neighbours. In our example, we provide a simple demonstration of the fact that when the k-value is increased and the convex hull becomes large, it tends to spread more and more away from the manifold. This analysis applies to all of the variations

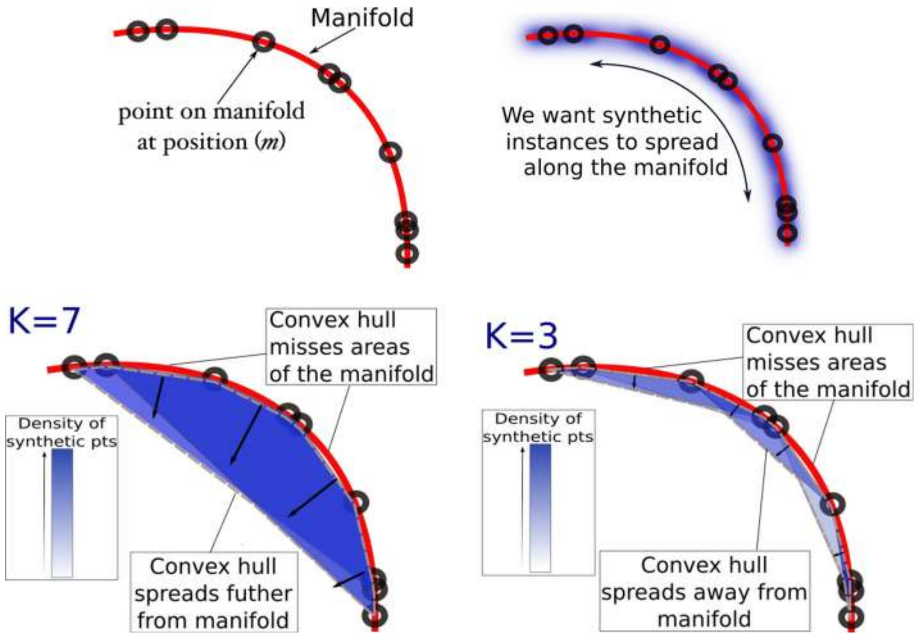


Fig. 2 Top left: One-dimensional non-linear manifold. Top right: spread of instances along the manifold. Bottom: spread of SMOTE-based synthesization away from the manifold with K = 7 and K = 3

of the SMOTE algorithm because the base algorithm does not account for geodesic distance along the manifold.

The weakness of SMOTE is due to the topological structure of a manifold; it will only produce an accurate kNN set if the query instances are close together (Gauld 2008). Similarly, a straight-line distance will only be accurate on a manifold when it is applied to measure the distance between instances in the same local neighbourhood. For this to occur, the training data must be dense and well sampled. When x_i and x_j are in different areas of the manifold, as we might expect on sparse imbalanced learning problems, the edge connecting them in the Hilbert space is likely to span regions of the data-space that do not belong to the minority class; however, this is precisely where SMOTE will generate instances!

4 Framework

4.1 Overview

Figure 1 presents the four components of our manifold-based synthetic oversampling framework. The framework that we proposed in our previous work has been extended here to first enable the evaluation of conformance to the manifold property.

Based on the design for the framework, we can assume that it will work well when the data conforms to the manifold property. However, it is often not clear if a domain conforms to the property. The first component of the framework assists in this determination. In addition, when it is established that the domain conforms to the manifold property, it offers the potential to empirically aid in the decision of which manifold learning method to utilize in the framework for the target data.

The determination of conformance to the manifold property has not been directly considered within the machine learning literature. We can, however, approach the question more generally from the perspective of the implicit dimensionality. This allows us to develop a conformance test that takes advantage of other areas of science, such as psychology and economics, that are interested in this question. The details of our proposed test are discussed in the subsequent section.

A large number of manifold learning methods, such as PCA, kernel PCA, autoencoding, local linear embedding, etc., have been studied within the machine learning literature. Individually, they incorporate a diverse set of assumptions and biases. For this reason, we have designed the framework that can take advantage of the wealth of manifold learning methods rather than designing a manifold-based synthetic oversampling algorithm that is restricted to a single manifold learning method. In the second element of the framework, the selected manifold learning method is formalized to induce a manifold representation of the minority class.

In general, the selection of a specific manifold learning method for a given problem will be guided by a number of factors. The most prominent amongst these are the number of training examples and the complexity of the latent manifold. If the learning objective involves a linear manifold, or the training data is extremely rare, a linear method is appropriate. Alternatively, non-linear problems with more training data are well-suited for methods that can represent the complexity.

In Sect. 4.3, we have selected PCA and DAE from the large number of manifold learning methods to demonstrate the framework. These two formalizations reinforce the steps required to implement the framework. In addition, they illustrate both a linear method and a non-linear method (PCA and DAE respectively). PCA is arguably the simplest, best known, and most widely applied manifold learning method. In addition, it can be seen to represent a reasonable baseline that more complex methods should beat, because it can be conceived as being at the origin of a line or perhaps the bottom of a partial order. Alternatively, we have selected DAE to demonstrate a non-linear formalization because its highly flexible structure renders it a reasonably generic method that can represent a wide variety of manifolds. Moreover, autoencoders offer the most natural means of sampling the induced manifold. Therefore, it is a good generic choice. However, in the spirit of our general framework, we encourage the consideration of alternative manifold learning methods.

Data is synthesized along the induced manifold during the second phase of the framework. For each manifold learning method, the details of the generation process will differ. The common objective, however, is to generate novel instances along the induced manifold to inflate the minority class. Our empirical results indicate that applying random transformations along the manifold works best.

We believe that random transformations produce a better synthetic set from the perspective of classifier induction because they reduce the likelihood that the samples are drawn from the extreme ends of the manifold. This advantage results from the fact that the generation process starts from points on the manifold where we have the most evidence that samples exist, and shift outward along the manifold to produce samples.

The final phase maps the synthesized data to the original feature space and returns it to the user. The reverse mapping is dependent on the selected algorithm. When selecting a manifold learning method, we must consider the ability to perform this mapping. It is more challenging, and perhaps not possible, for some approaches. Both PCA and DAE offer straightforward means for reversing the mapping.

4.2 Testing conformance to the manifold property

The flexibility of the proposed framework enables it to be widely applicable to datasets with various degrees of conformance to the manifold property. Nonetheless, the performance is expected to increase with the conformance to the property. Thus, it is desirable to have a metric to test for conformance when selecting between manifold-based synthetic oversampling and SMOTE. Typically when deciding whether or not to use manifold learning, researchers have relied on their experience and intuition. A key component of this comes from the expectation that high-dimensional domains are likely to conform to the manifold property. Part of our research is to examine more quantitative approaches.

We propose that a metric for conformance to the manifold property should have the following attributes:

- It should produce a continuous score that can be used to rank datasets; and,
- Given an independent random sample of datasets, the corresponding ranks should be approximately uniformly distributed so that datasets with various levels of conformance are accurately and evenly spread over the range of scores.

With a metric of this nature, we can experimentally select a threshold below which manifold-based synthetic oversampling learning should be applied. We look to the established field of factor analysis in order to develop this metric. Factor analysis is a toolbox of methods for estimating the implicit dimensionality of a dataset. Given a factor analysis method $\mathcal{F}(\cdot)$ and a dataset D, an integer y is returned that indicates the implicit dimensionality as $y = \mathcal{F}(D)$.

Conformance to the manifold property assumes that the probability density of the data distribution resides in a lower dimensional space. Therefore, we can say that conformance to the manifold property increases as implicit dimensionality decreases relative to the corresponding feature space. Based on this, we measure conformance to the manifold property as:

$$m(D) = \mathcal{F}(D)/dim(D). \tag{2}$$

We use the ratio of the implicit over the actual dimensionality of the dataset to give preference to datasets having a relatively smaller implicit dimensionality.

Factor analysis methods are particularly popular in the social sciences where the latent variables associated with the topic are either unknown or hard to measure directly. An important question from the field is how to best estimate the number of factors to retain (Courtney and Ray 2013). The methods are generally based on principle component analysis and principle factor analysis, and the insight of Cattell (1966) that eigenvalues, when plotted in descending order of magnitude against their factors, 'level off' at the point where the factors are primarily measuring random noise. Thus, the general goal is to find the elbow in the curve so as to keep all of the components that are not associated with random noise. In our terms, those above the elbow can be thought of as representing the manifold.

The traditional approaches to answering this question utilize the Kaiser criterion and the scree test (Kaiser 1960; Cattell 1966). Given the practical importance of factor analysis to the field, the research continues to evolve. Simulation studies have shown that methods such as parallel analysis (Horn 1965), minimum average partial procedure (Garrido et al. 2011) and comparison data (Ruscio and Roche 2012) can be more accurate than the scree test and Kaiser criterion. Within machine learning, Zhu and Ghodsi (2006) proposed an alternate means of estimating the implicit dimensionality using the profile likelihood.

In the 1950s, Henry Kaiser famously proclaimed, "solving the number of factors problem is easy, I do it everyday before breakfast. But knowing the right solution is harder." He was recognizing that even at that early stage, there were a wide variety of methods available

Table 1 Factor analysis methods

Acronym	Summary	References
PL	Profile likelihood: searches for the scree by finding the λ_n that maximizes the difference between the distribution of $1 \ldots n$ and $n + 1 \ldots m$, where n is the number of eigenvalues	Zhu and Ghodsi (2006)
Fact	Factors: compares the scree of factors of the observed data with that of a random data matrix. Reports the number of factors with eigenvalues > eigenvalues of random data	Revelle (2013)
Comp	Components: compares the scree of components of the observed data with that of a random data matrix. Reports the number of components with eigenvalues > eigenvalues of random data	Revelle (2013)
MAP	Velicer's minimum average partial criterion: applies principal components analysis and follows this by examining a series of matrices of partial correlations	Revelle and Rocklin (1979)
VSS	Very simple structure criterion: compares the original correlation matrix to that reproduced by a simplified version of the original factor matrix	Velicer (1976)
BIC	Bayesian information criterion: chooses the most likely model from a set of models	Schwarz (1978)
ABIC	Sample size adjusted BIC: chooses the most likely model from a set of models	Schwarz (1978)
PA	Parallel analysis: creates a random data matrix and compares the eigenvalues values calculated on it to the eigenvalue calculated on the target domain. All components with eigenvalues greater than the mean of the eigenvalues for the random data are kept	Humphreys and Montanelli (1975)
CD	Data comparison: variant on PA that reproduces the observed correlation matrix rather than generating random data	Ruscio and Roche (2012)
$\lambda > \mu$	$\lambda > mean(\lambda)$: selects the end of the scree as the point where the eigenvalues become less than the mean of the eigenvalues	Revelle (2013)
OC	Optimal coordinate: determines the location of the scree by measuring the gradients associated with eigenvalues and their preceding coordinates	Raiche et al. (2006)
AF	Acceleration factor: numerical solution for determining the coordinate where the slope of the curve changes most abruptly	Raiche et al. (2006)

and no single method seemed to offer a clear advantage for all datasets and all objectives. Rather, the choice of a method is largely experimental that depends on the domain and the overarching objective of the research. For this reason, we evaluate a wide variety of methods with various degrees of complexity in order to empirically select the most suitable approach for our application. These are listed in Table 1.

4.3 Instantiations

4.3.1 Instantiations with PCA

PCA is a linear mapping from the d-dimensional input space to a k-dimensional embedded space where $k \ll d$. The standard process is a result of calculating the leading eigenvectors E corresponding to the k largest eigenvalues λ from the sample covariance matrix Σ of the target data.

In the PCA realization of the framework, a model $pca = \{\mu, \Sigma, E, \lambda\}$ of the d-dimensional target class T with m instances is produced. We produce a synthetic set S of n instances in the manifold-space by randomly sampling n instances from $T' = T \times E$ (T in the PCA-space) with replacement. In order to produce unique samples on the manifold, we apply *i.i.d.* additive Gaussian noise $\mathcal{N}\big(0, \mathcal{D}\big)$ to each sampled instance prior to adding it to the synthetic set S. The covariance matrix for the Gaussian noise is a diagonal matrix with each $\sigma_{i,i}$ specified by $\beta\lambda_i$, where β is the scaling factor applied to the eigenvalues. This controls the spread of the synthetic instances relative to the manifold, and can be thought of as a geometric transformation of points along the manifold, thereby producing new synthetic samples on the manifold. Finally, we map the synthetic instances S into the feature space as $S' = S \times E^{-1}$ and return them to the user for use in classifier induction.

4.3.2 Instantiations with autoencoders

Autoencoders are a form of artificial neural networks commonly used in one-class classification (Rumelhart et al. 1986; Japkowicz 2001). They have an input layer, a hidden layer and an output layer, with each layer connected to the next via a set of weight vectors and a bias. The input and output layers have a number of units which is equal to the dimensionality of the target domain, and the user specifies an alternate dimensionality for the hidden space. The learning process involves optimizing the weights used to map feature vectors from the target class into the hidden space, and those used to map the data from the hidden space back to the output space.

A manifold bias is incorporated in the autoencoding process through its mapping from the feature space to the hidden-space and back via $f_\theta(\cdot)$ and $g_{\theta'}(\cdot)$, where:

$$\begin{aligned} f_\theta(x) &= s(\mathbf{W}x + b) \\ g_{\theta'}(y) &= s'(\mathbf{W}'y + b'). \end{aligned} \tag{3}$$

Here, x is a d-dimensional input vector, and θ and θ' represent the induced encoding and decoding parameter set, respectively. Specifically, $\mathbf{W}$ is a $d \times d'$ weight matrix and b is a d'-dimensional bias vector, where d' is the number of hidden units. The function s is a non-linear squashing function, such as the sigmoidal. In the decoding parameter set, $\mathbf{W}'$ 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. As is standard with artificial neural networks, the weights are learnt using backpropagation and gradient descent. In addition, we utilize denoising with additive Gaussian noise during the training process as a form of regularization to promote the learning of key aspects of the input distribution (Vincent 2010). We add Gaussian noise to the input and the network learns to reconstruct the clean instances.

The learning processes prioritizes the dual objective of a reconstruction function $g(f(\cdot))$ that is as simple as possible, but capable of accurately representing neighbouring instances from the high-density manifold (Alain and Bengio 2014). This promotes accurate reconstruction of points on the manifold, whilst the reconstruction error $|x - g(f(x))|^2$ rises quickly for examples orthogonal to the manifold. Given a point, p, on the manifold, the output $g(f(p))$ remains on the manifold in essentially the same location. Conversely, when an arbitrary point, q, is sampled from off the manifold, the output $f(q)$ is mapped orthogonally to the manifold. Therefore, $g(f(q))$ returns a representation of q on the high-density manifold. This is the inspiration for the DAE sampling method, and is demonstrated in Step

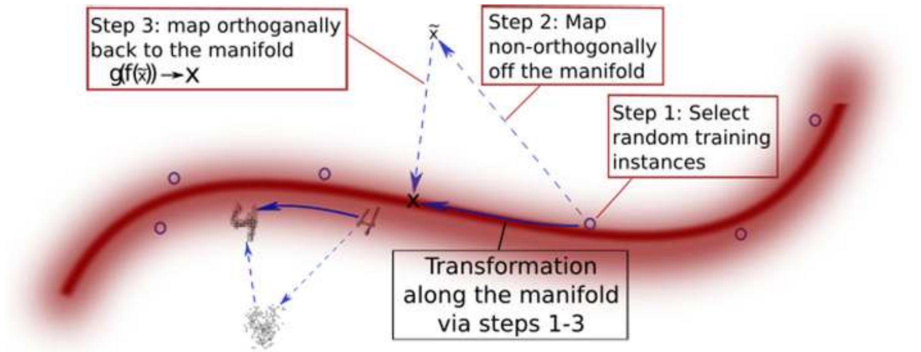

Step 3: map orthoganally back to the manifold
g(f(x̃))→X
x̃
Step 2: Map non-orthogonally off the manifold
Step 1: Select random training instances
X
Transformation along the manifold via steps 1-3

Fig. 3 Three steps of synthesization for the autoencoder formalization with generic points and handwritten 4s

3 of Fig. 3 as $g(f(\tilde{x})) \rightarrow x$, where $\tilde{x}$ is a point off the manifold, with the manifold depicted in red.

Whilst the encoding-decoding function $g(f(\cdot))$ is typically considered as a single unit, for the purpose of our framework they are treated separately. The encoding induces the lower-dimensional manifold representation and is used to generate samples and the decoding function performs the reverse mapping. Specifically, we take an arbitrary minority class instances x and apply a non-orthogonal mapping. This shifts it off the implicit manifold $x \rightarrow \tilde{x}$. It is then orthogonally mapped back to the manifold via $f(\tilde{x}) \rightarrow y$. The result is a transformation along the implicit high-density manifold from a training instance x to a synthetic instance y. This is illustrated graphically in steps 1–3 of Fig. 3. The non-orthogonal mapping is produced by adding noise to the training instance x. A greater amount of noise leads to a larger transformation along the manifold. By sampling n instances from the minority class with replacement and performing the random transformation, we produce the synthetic set. The decoding function $g(\cdot)$ maps the synthetic instances on the manifold in the hidden space to their corresponding points on the implicit manifold in the feature space. Algorithm 1 formalizes the method.

```
Algorithm 1 dae-SyntheticOversampling($\mathcal{X}$, $DAE_{\{\mathbf{W},b\}}$, $n$, $\sigma$)
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) $\sigma$, variance of the Gaussian sample initiation noise.
Output:
    i) $\mathcal{Y}$, the synthetic samples.
Method:
$\mathcal{X}'$: column normalization of $\mathcal{X}$ between $[-1, 1]$.
$normParams$: column normalization parameters of $\mathcal{X}$.
$\mathcal{Z}$: normalized $\mathcal{X}$ plus sample initiation noise $\mathcal{N}(0, \sigma)$.
$\mathcal{Y}' = DAE_{\{\mathbf{W},b\}}(Z)$: samples $\mathcal{Y}'$ from the induced manifold.
$\mathcal{Y}$: denormalization of $\mathcal{Y}$ based on $normParams$.
$Return(\mathcal{Y})$
End Algorithm
```

Parameter selection is an important task in machine learning. Prior to calling Algorithm 1, we perform model selection by randomly searching the parameter space for a predetermined number of iterations with the objective of minimizing the reconstruction error on the minority training data $\mathcal{X}$. This is simple and efficient, and it alleviate us of the task of hand tuning the model. The ability to do this is a very nice feature of PCA and autoencoders. This method works very well, as is demonstrated by our results.

The size of the parameter search space should be kept small by limiting the upper bound of the epochs and number of hidden units. We do this to limit the complexity of the model and avoid overfitting the small amount of training data that is available. For the 250-dimensional spectra data, we searched 5–30 hidden units with fewer than a thousand epochs of training. Indeed, overfitting is the primary risk of poorly selected parameters. In terms of manifold learning on sparse data, this is realized in the form of an induced manifold with wildly unrealistic curvature. As we have specified, however, denoising during training and limiting the upper bound of the hidden space in the autoencoder minimizes the risk of overfitting.

5 Demonstration

In this section, we visualize the results of both manifold-based synthetic oversampling and non-manifold-based synthetic oversampling on four domains that are commonly utilized in the manifold learning literature. In particular, the demonstrations in the following two subsections utilize theoretical manifold distributions involving points on a three-dimensional helix and Swiss roll (Xue and Chen 2007; Wei et al. 2008; Goldberg et al. 2009; Silva and Tenenbaum 2003; Weinberger et al. 2004a). These lead into two subsections with more realistic image classification domains. Specifically, the third and fourth experiments utilize high-dimensional datasets composed of images of a rotating teapot and of handwritten digits (Weinberger et al. 2004b; LeCun et al. 1998).

The objective of these experiments is to demonstrate that non-manifold-based synthetic oversampling has an elevated risk of generating instances in low probability regions of the data space that are orthogonal to the embedded manifold, whilst manifold-based approaches spread the synthetic instances along the manifold. The weaknesses of the non-manifold-based approaches result from the error-prone means by which they identify nearby instances. Specifically, the failure to account for the geodesic distance between points (Belkin and Niyogi 2004). The implications of this are presented pictographically in Fig. 4. Here, the

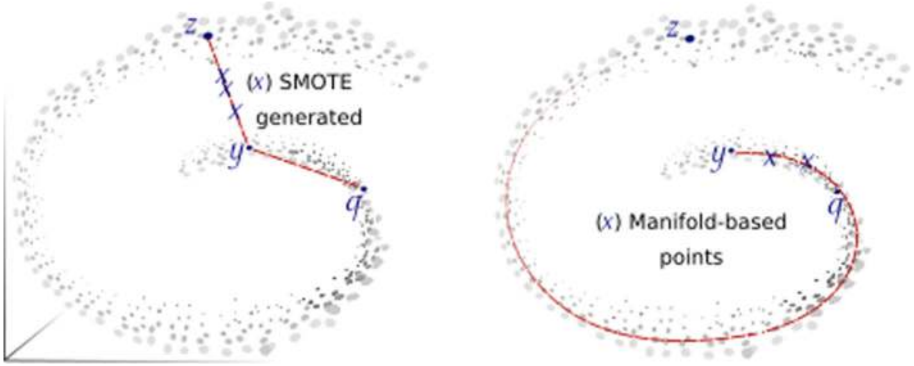


Fig. 4 (Left) SMOTE styled generation that ignores the geodesic distance. (Right) Generation along the manifold between geodesically near instances

distances measured on the planes spanning points (y, z) and points (y, q) are the same. Thus, to SMOTE these points are equivalently valid candidates from which to generate synthetic training instances. When observed with respect to the curvature of the manifold, however, q is significantly closer than y. It is erroneous equivalences such as these that cause non-manifold-based methods to synthesize instances in low probability regions that are orthogonal to the manifold.

For each demonstration, we use 25 random samples to train and generate novel synthetic instance from. The denoising autoencoder and PCA implementations of the manifold-based synthetic oversampling framework are employed. In addition, we demonstrate the generative capabilities of SMOTE and kernel-based synthetic oversampling. The kernel-based method divides the data space into hypercubes and uses Parzen-windows to estimate the density in the hypercubes. These densities are smoothed using a Gaussian kernel. Although, kernel-based methods are not widely applied, they have seen some attention (Gao et al. 2012).

To summarize, each subsection below takes a dataset that conforms to the manifold property and demonstrates that the existing approaches generate instances that spread away from the latent embedded density, whilst our proposed method generates instances that spread along the embedded structure. Based on this, we claim that manifold-based synthetic oversampling has the best chance of producing instances to improve the performance of the target classifier. This latter result is shown in Sect. 7.

5.1 Helix

The helix distribution is consistently utilized for demonstration purposes in manifold learning. In order to increase the difficulty of the modelling task, and to 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 employed. 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 5 plots the training data and the synthetic data produced by the denoising autoencoder and PCA formalizations of the framework along with SMOTE and the kernel-based method. The circles in the figure represent the training data and the triangles are the instances synthesized by each method; the colouration of the points is simply to emphasize the relative positions. The red boxes highlight the weaknesses of each method.

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. 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, the denoising autoencoder generates a sprinkling of instances along the manifold, much like a Canadian snowplow traveling 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.

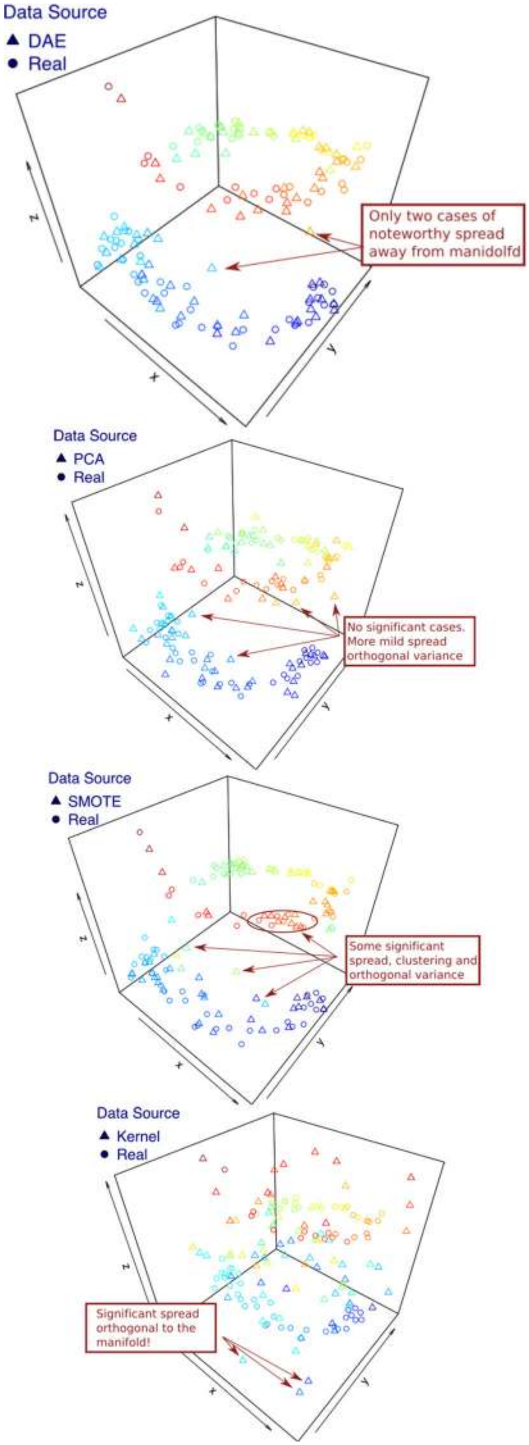


Fig. 5 From top to bottom, helix data synthesized by DAE, PCA, SMOTE and kernel-based methods. The colouration in these plots merely provides perspective on the relative distance between points. The shape of the point specify whether it is a synthesize instance (triangle) or real instances (circle)

5.2 Swiss roll

Like the helix, the swiss roll is a common dataset in the manifold learning literature. 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 are presented in Fig. 6; 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 part of the swiss roll. Three instances are, for example, synthesized in the void that is the center 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 data space 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.

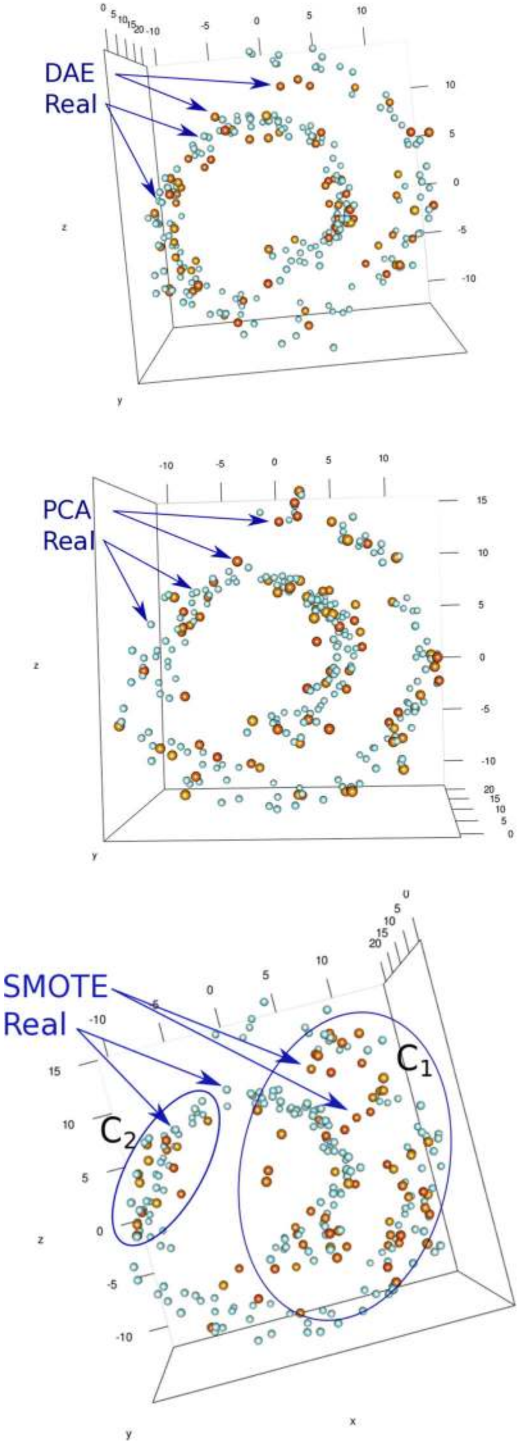
When we take a macro-scale view of the synthesized data, we see that instead of a swiss roll, SMOTE has produced two clusters of synthetic instances. These are circled and marked by C_1 and C_2 in the figure. It generated a small cluster on the left of the data space. This cluster represents a small area of the swiss roll. The other cluster is much larger in terms of the area of the 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.

5.3 Handwritten fours

Handwritten 4s from the *MNIST* 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. This is the space that encodes the various, legitimate shifts, rotations and skews in the target digit. In this space, we are much more likely to synthesize fours correctly.

A random selection of 16 fours that were generated by each synthetic oversampling method are presented in Fig. 7. This includes the generators of primary interest; manifold-based synthetic oversampling framework using the denoising autoencoder and PCA instantiations, SMOTE and kernel-based oversampling. 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 clearly shown to be 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 that modelling and synthesization is performed in the fog of the feature space rather than in the clarity of the manifold space.

Fig. 6 From top to bottom, swiss roll data synthesized by denoising autoencoder (DAE), PCA SMOTE-based methods



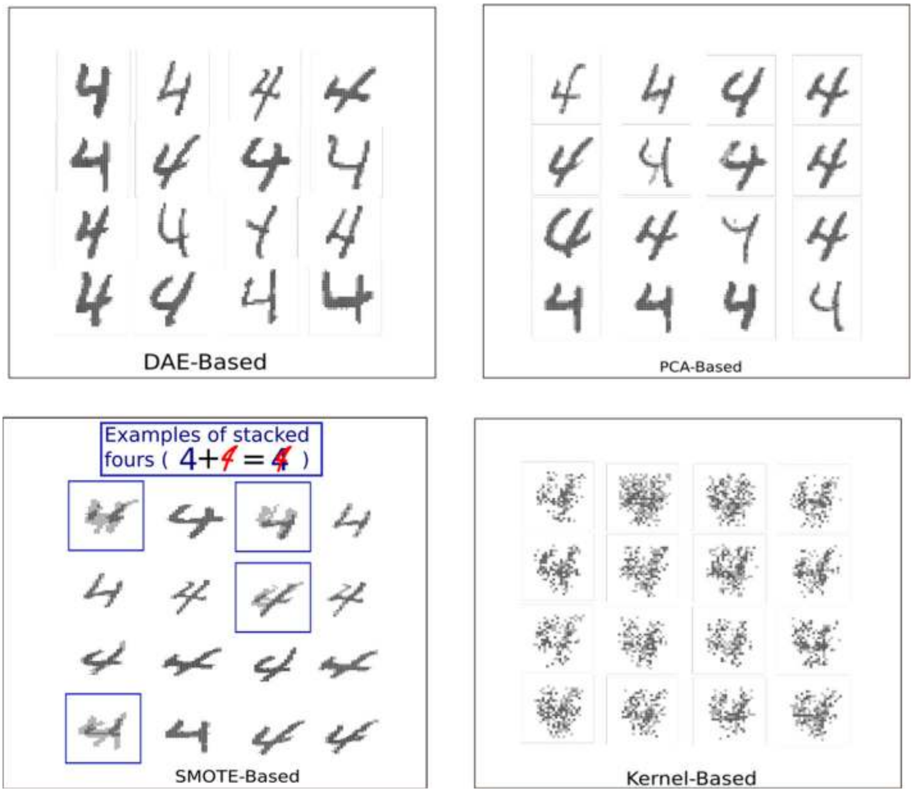


Fig. 7 From left to right, handwritten fours synthesized by DAE, PCA, SMOTE and kernel-based methods

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. These fours occur when SMOTE generates new instances between two training instances that are far apart. The vector connecting the two training instances along which SMOTE produces synthetic instances does not follow the curvature of the manifold. The longer the vector connecting the nearest neighbours, the more likely it is that the vector will deviate from the manifold. Therefore, the synthesized instances along this vector are not accurate 4s.

PCA produces reasonable 4s. Only three of the examples are of lesser quality and none are skewed in any way. 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.

5.4 Rotating teapot

The final demonstration domain involves the teapot dataset. This dataset includes 400 colour images of a teapot from different angles; thus, the dataset appears as 400 snapshots of a rotating teapot. The images have a resolution of 76 × 101 with each pixel involving 3 bytes

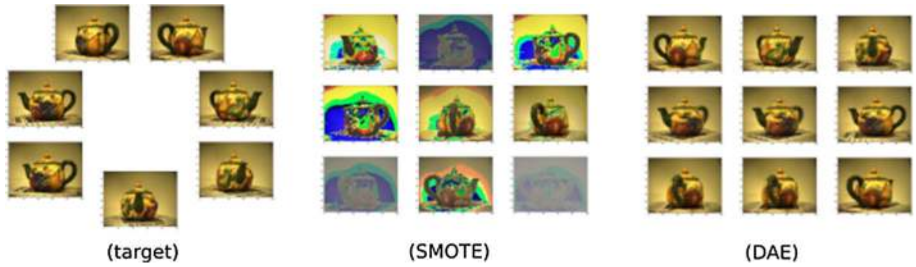


Fig. 8 (Left) A subset of the 30 training instances. (Centre) SMOTE generated instances. (Right) Instance generated via manifold-based oversampling

of colour information. Therefore, the dataset has 400 instances with 23,028 dimensions. For training and synthetic generation purposes, 30 instances are taken at random. Given the high-dimensionality, this is, indeed, a very small training set. In spite of the high-dimensionality, however, this domain can effectively be represented with as little as a single degree of freedom that accounts for the angle of rotation (Silva and Tenenbaum 2003). To this end, a synthetic oversampling method that utilizes a manifold-based approach is expected to accurately synthesize new instances even with a relatively small training set.

For this final demonstration we focus our attention on the DAE and SMOTE algorithms alone as they have consistently been the best methods from the manifold-based and non-manifold-based camps. A subset of the training instances along with instances generated with the SMOTE and DAE algorithms are displayed in Fig. 8. The generated results on the teapot dataset profoundly demonstrate the limitations of SMOTE on a high-dimensional domain that conforms to the manifold property. Moreover, these results illustrate the effectiveness of manifold based oversampling.

6 Experimental method

In order to validate the benefit of manifold-based synthetic oversampling, we apply it to a wide variety of datasets with a diverse set of classifiers. The details are presented in the following sections.

6.1 Datasets

6.1.1 Gamma-ray spectra

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) and the monitoring of ports of entry for the importation of illicit nuclear materials (Kangas et al. 2008). 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

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

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 within 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 of isotopes of interest. The work in this article 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 and Development Canada, Canadian Security Intelligence Service, etc., to deploy 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 spectra for isotopes of interest that may signify a person transporting a material that poses a radioactive threat to participants and spectators at 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 (Vancouver). 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).

The Vancouver dataset has 512-dimensions and was recorded in one-minute samples. This produced gamma-ray spectra that are very noisy. The data is composed of pure background readings and a background plus isotopes of interest. The latter forms the minority class in our experiments. There are 39,000 background instances and 39 minority class instances involving Iodine, Thallium, Technicium and Caesium.

The environmental monitoring data is collected in fifteen-minute samples by the national monitoring network of gamma-ray spectrometers. 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. We use data from two locations in our experiments (Thunder Bay and Saanich). The Saanich, BC dataset has 19,068 background readings and 44 isotopes of interest. The Thunder Bay, Ontario dataset contains 11,573 background instances and 29 isotopes of interest. In each of these datasets, and the Vancouver dataset, the classification task is made more difficult by the complexity of the background, the decay of the isotope and the presence of heavy rain. Each of these obscures the isotopes signature in the measured spectra.

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

Table 2 UCI Datasets applied in these experiments

	Min class	Dim
Breast	Malignant	9
Diabetes	Positive	8
Ecoli	1	7
Heart Statlog	Present	13
Ionosphere	B	34
Letter	R	16
Musk	1	166
Opt Digits	4	64
Ozone One	1	72
Pen Digits	3	16
Satlog	4	36
Segmentation	Brickface	19
Sonar	Rock	60
Vehicle	Saab	18
Wave	1	40
Yeast	MIT	8
CIFAR-10	Horse	1024

6.1.2 Benchmark domains

The sixteen UCI datasets along with the high-dimensional image classification domain, CIFAR-10, are presented in these experiments. These domains are specified in the first column of Table 2. The UCI datasets were selected to ensure a diverse range of dimensionalities and complexities. When required, the datasets are converted to a binary task by selecting a single class to form the minority class, and the remaining classes are merged into one.

The CIFAR-10 dataset is a high-dimensional image classification dataset that appears frequently in the applied manifold and deep learning literature (Krizhevsky 2009). The dataset includes 60,000 32 × 32 pixel colour images. The dataset has 10 classes associated with different objects and animals. For the purpose of our experiments, we have artificially undersampled the horse class to render it a minority class. We selected the horse class because the images are somewhat similar to the deer, dog and cat classes. Thus, the modified classification task is of high complexity.

For each experiment, we train on 25 minority training instances and 250 majority training instances; thus, we render each domain as an absolutely imbalanced classification task. We have selected constant values for the training distribution of each dataset, rather then specifying a percentage for the minority class, in order to ensure that the performance differences between datasets are not the result of having access to different numbers of minority instances. If we set the minority portion to 10%, for example, then a dataset with 1000 instances would have many more examples in the training set than a dataset with 200 instances.

6.2 Algorithms and evaluation

We utilize the SVM, MLP, kNN, naïve Bayes and decision tree classifiers in the following experiments. Synthetic oversampling is performed by the autoencoder and PCA instantiations of our framework. These are compared to random oversampling (ROS), SMOTE, Borderline SMOTE and SMOTE with the removal of Tomek links. For each variation of the SMOTE algorithm, we test k-nearest neighbour with $k \in \{3, 5, 7\}$.

With respect to the gamma-ray spectra classification results in Sect. 7.1, we report the mean five times twofold cross-validated AUCs. Five times twofold cross validation is used in place of the more common tenfold version because it has been observed that it has a lower probability of issuing a Type I error (Dietterich 1998). In addition, k-fold cross validation with larger k values was established with small datasets in mind; the size of the datasets is not a concern here. Because the classification objective is to select the best synthetic oversampling method for each dataset, here we test for statistical significance of using manifold-based synthetic oversampling versus SMOTE using the t test.

In order to further validated manifold-based synthetic oversampling and illustrate the impacts of the manifold property on the standard methods of synthetic oversampling, we perform an additional set of experiments on the benchmark datasets. We apply a different experimental setup here as the datasets are diverse and our objectives are different.

In order to limit the impact of sampling during the preprocessing to create imbalanced binary classification problems, we record mean AUC performance results over thirty trials for the baseline classifiers and the classifiers aided by each synthetic oversampling method. Given the large number of datasets and algorithms, we test our results for statistical significance using the Friedman test and Nemenyi post-hoc test.

7 Results

7.1 Gamma-ray spectra

The results for the experiments on the Saanich, Thunder Bay and Vancouver dataset are presented in Fig. 9. In each experiment, we report the mean AUC produced by the individual classifiers with synthetic oversampling via the manifold method (DAE or PCA) and the SMOTE-based methods.

The general results based on the application of the five classifiers on the three gamma-ray spectra datasets demonstrate that both manifold-based oversampling and the SMOTE-based algorithms are always better than random oversampling. Moreover, manifold-based synthetic oversampling is nearly always better than the SMOTE-based algorithms.

Over the three datasets and five classifiers, manifold-based synthetic oversampling produces a greater improvement on 13 of the 15 cases. This shows that it has a stronger positive impact than SMOTE on a wide range of classifiers. Moreover, the combination of manifold-based synthetic oversampling with a classifier produces the best overall mean AUC for each of the three dataset. This occurs with Naïve Bayes on Saanich, MLP on Thunder Bay and kNN on Vancouver.

To summary, when applied to the gamma-ray spectral classification datasets, manifold-based synthetic oversampling has a better impact on individual classifiers in general. More importantly, it produced the best overall classifier in each case.

7.2 Benchmark datasets

In order to generalize our findings, we now shift to examine the impact of the manifold on synthetic oversampling over benchmark datasets from the UCI repository and the CIFAR-10 image classification dataset. The mean of the AUC results are tabulated in Table 3.

Fig. 9 AUC results on the gamma-ray spectra datasets broken down by classifier. The plots listed from top to bottom are for: Saanich, Thunder Bay and Vancouver

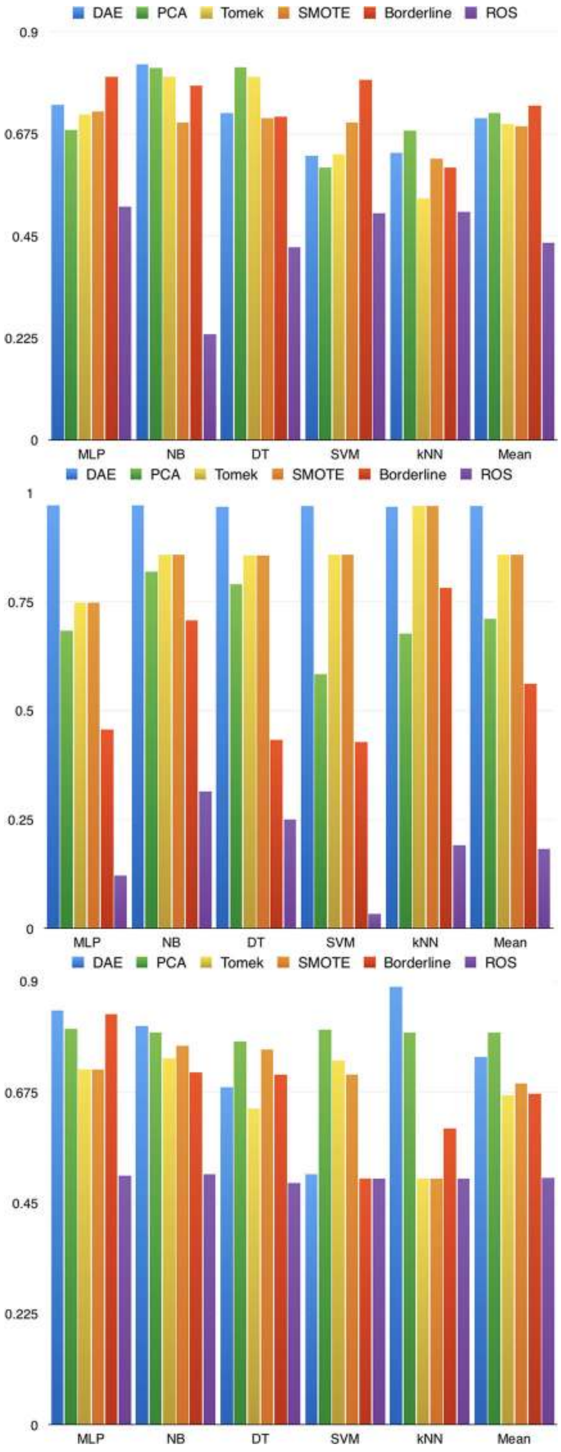


Table 3 AUC results on the benchmark datasets for the baseline, SMOTE, borderline SMOTE, SMOTE with the removal of Tomek links and the two manifold-based methods PCA and DAE

	Base	SMOTE	BLS	Tomek	PCA	DAE
Musk2	0.724	0.776	0.773	0.431	0.767	**0.793**
Opt Digits	0.828	0.830	0.834	0.581	0.850	**0.916**
Wave-form	0.657	0.725	0.735	0.547	0.733	**0.755**
Satlog	0.675	0.770	0.764	0.545	0.776	**0.778**
Ionospher	0.778	0.831	0.831	0.489	**0.836**	0.829
Sonar	0.724	0.733	0.729	0.496	**0.742**	0.740
Ozone One	0.625	0.710	**0.711**	0.561	0.709	0.702
Segment	0.864	0.889	0.895	0.541	0.879	**0.960**
Vehicle	0.581	0.657	0.656	0.445	0.665	**0.667**
Pen Digits	0.946	0.957	0.742	0.960	0.972	**0.974**
Breast	0.915	0.930	0.950	0.742	0.943	**0.953**
Yeast	0.602	0.703	0.705	0.539	**0.707**	0.653
Ecoli	0.887	0.937	0.710	0.509	**0.950**	0.923
Heart	0.755	**0.782**	0.764	0.627	0.776	0.770
Letter	0.762	**0.936**	0.764	0.545	0.878	0.870
Diabetes	0.569	**0.709**	0.637	0.445	0.662	0.652
CIFAR-10	0.528	0.522	0.529	0.510	0.533	**0.555**
Total Wins	0	3	1	0	4	9

Bold values indicate the highest AUC for each data set

The results show that performing synthetic oversampling prior to classifier induction improves the AUC performance beyond the baseline on every dataset. Moreover, the benefit of synthetic oversampling is large (greater than 0.1 AUC) in many cases, such as on the optical digits, letter, yeast and diabetes datasets.

We use the Friedman test to evaluate the statistical significance in the performance of each method. The null hypothesis states that there is no difference between the methods over all datasets. These results show a statistically significant difference in the performance of the methods with a p value of 2.213e$-$06. The Nemenyi multiple comparison test enables us to identify where the differences exist. It indicates that each synthetic oversampling method is statistically different than the baseline.

When we shift to examine the relative performance of the synthetic oversampling methods, we see that manifold-based synthetic oversampling generally produces the largest increase over the baseline. This occurs 13 out of the 17 times.

Figure 10 displays relative performance of manifold-based synthetic oversampling to the SMOTE-based methods. In producing these results, we have selected the best manifold-based method (MOS_{PCA} or MOS_{DAE}) for each dataset from Table 2, and have done likewise for the SMOTE-based methods. The results, $AUC(MOS(D)-AUC(SMOTE(D)))$, are plotted in decreasing order of the dimensionality of the datasets. This reveals the trend of superiority for manifold-based synthetic oversampling on higher dimensional datasets. This suggests that dimensionality can be used to determine when it is best to apply manifold-based synthetic oversampling. Whilst dimensionality is often used as a proxy for conformance to the manifold property, we recognize that conformance is much more complex. In the next section, we explore the means introduced in Sect. 4.2 as better methods of assessing conformance to the manifold property.

An additional observation from Table 3 is that there is a clear distinction between the datasets that are well suited for MOS_{DAE} and those that are better suited for MOS_{PCA}.

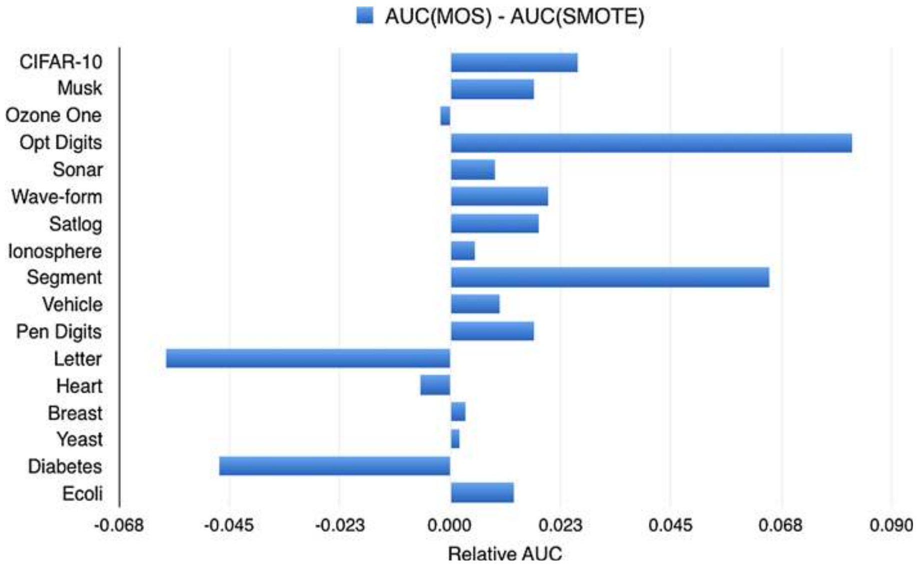


Fig. 10 Bar graphs displaying the relative difference in AUC between the best manifold-based synthetic oversampling (MOS) method to the best SMOTE-based approach. The results are sorted by data dimensionality in descending order

Specifically, on certain datasets, such as Ionosphere, Sonar and Ozone One, where the performance of MOS_{DAE} drops below SMOTE, the performance of MOS_{PCA} becomes better than MOS_{DAE}.

On some datasets, even though MOS_{PCA} is the better of the two formalizations of the framework that were considered, it is not better than SMOTE. Here, the utilization of a third manifold method might be more suitable. This is indicative of the benefit of developing a framework that enables the user to select the manifold learning method that has the most appropriate bias for the learning problem.

These results demonstrate that using manifold-based synthetic oversampling is generally preferable to the application of SMOTE-based methods, and that the advantage increases with dimensionality. Whilst the relationship between dimensionality and conformance to the manifold property is often assumed, dimensionality is not necessarily the most accurate metric for use in deciding which synthetic oversampling method to use. We are interested in formalizing the relationship between the choice of synthetic oversampling algorithm and the data. We examine this question in the following section.

7.2.1 Manifold conformance and the loss metric

In this section, we test our function for estimating conformance to the manifold property. The objective is to define a function to assess which synthetic oversampling algorithm to apply to a given dataset. The specific function was introduced as $m(D) = \mathcal{F}(D)/dim(D)$ in Eq. 2 in Sect. 4.2. It ranks machine learning datasets according to their conformance to the manifold property. In order to do this, it reports the ratio between the implicit dimensionality of the data, as measured by a user-specified function, and the physical dimensionality of the data.

As discussed in Sect. 4, there has been very little work within machine learning aimed at assessing the implicit dimensionality of a dataset. Outside of machine learning, however,

Table 4 Correlation between difference $AUC(MOS)-AUC(SMOTE)$ and $m_{\mathcal{F}}(\cdot)$

	COMP	FACT	MAP	BIC	ABIC	DC	$\lambda > \mu$
DIFF	−0.51	−0.51	0.35	−0.14	−0.26	0.38	−0.01

	PA	OC	AF	PL	Dim
DIFF	−0.45	−0.38	−0.52	−0.31	0.14

this is a thriving area of study. This work reports this first large scale study of these methods from the perspective of manifold learning in a machine learning context.

We tested each function $\mathcal{F}$ in the set of methods discussed in Sect. 4 to evaluate their effectiveness in the machine learning context. Moreover, we compared them to the standard, by a more naïve, approach of assuming higher dimensional domains have greater conformance. Our hypothesis is that selecting a good $\mathcal{F}(\cdot)$ methods for use in $m_{\mathcal{F}}(\cdot)$[3] will produce a better ranking of the datasets than dimensionality alone.

We use the UCI datasets and augmentation versions of the UCI datasets for these experiments. The augmented versions increase the dimensionality of the data to embedded its probability density in a lower space; this process is described in Bellinger (2016). Incorporating the augmented versions increases the sample size and the variance in the conformance to the manifold property to produce a stronger assessment of the $m(\cdot)$ score.

For evaluation, we compare the relationship between $m_{\mathcal{F}_i}(D_j)$ and $dim(D_j)$ to $AUC(MOS(D_j))-AUC(SMOTE(D_j))$ for each dataset D_j. For a good choice of $m_{\mathcal{F}}(\cdot)$, we expect a negative correlation. This is because lower implicit dimensionality causes a lower score, and the relative performance of manifold-based synthetic oversampling increases with greater conformance. For $dim(D_j)$, we expect a positive correlation because higher dimensionality serves as a surrogate for greater conformance to the manifold property.

The primary question is, do the functions $\mathcal{F}_i$, produce a better ranking than dimensionality alone? We assessed this using correlation analysis with linear regression over $m_{\mathcal{F}_i}(D_j)$ and $AUC(MOS(D_j))-AUC(SMOTE(D_j))$, and $dim(D_j)$ and $AUC(MOS(D_j))-AUC(SMOTE(D_j))$. The correlation results are reported in Table 4. From this, we see that a few of the standard methods have a stronger correlation then dimensionality. The $m_{\mathcal{F}}$ score using $\mathcal{F} \in \{COMP, FACT\}$ have the strongest correlation with the relative performance.

Figure 11 plots the performance differences between manifold-based synthetic oversampling and SMOTE. The datasets are sorted according to their $m_{\mathcal{F}_i}(\cdot)$ score using COMP. The datasets at the top of the plot have the greatest conformance to the manifold property and those at the bottom have weak conformance. Manifold-based synthetic oversampling produces better performance on roughly the top two-thirds of the datasets. This sorting suggests that the $m_{\mathcal{F}_i}(D_j)$ using COMP works quite well. FACT, the other method with a strong correlation, produces a similar sorting, whereas using $dim(D_j)$ is more mixed as suggested by the correlation. We have omitted their bar graphs in the interest of brevity.

Using the ranking produced by $m_{\mathcal{F}_i}(\cdot)$ score, we can sort the datasets in a manner related to the performance of the synthetic oversampling methods. This enables the setting of threshold a τ to dichotomize datasets such that for $\forall D : m(D) < \tau$, manifold-based synthetic oversampling should be applied. Table 5 illustrates this for $m_{\mathcal{F}_i}(\cdot)$ score using $COMP$. It places 21 datasets below the threshold; manifold-based oversampling outperforms SMOTE

[3] Note that we have added the subscript $\mathcal{F}$ to the $m(\cdot)$ score to signify the use of a specific function Eq. 2.

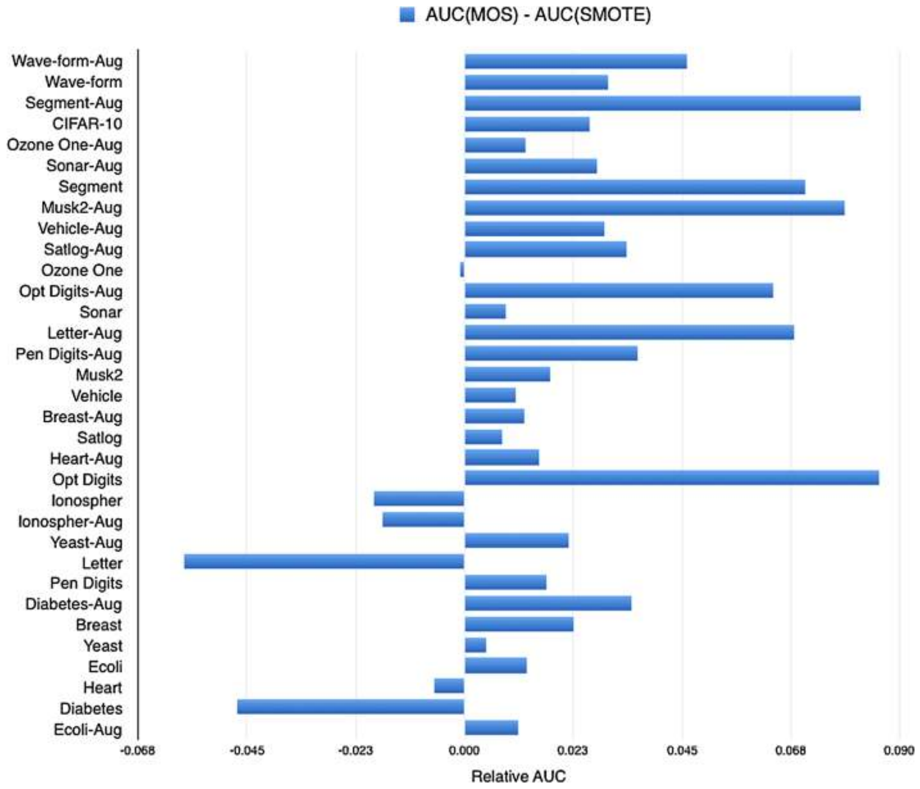


Fig. 11 Bar plots of the performance difference $AUC(MOS){-}AUC(SMOTE)$ sorted by the top $M(\cdot)$ methods

Table 5 Number of times each method produced the top mean AUC above and below the $m_{\mathcal{F}_i}(\cdot)$ score threshold using $COMP$ in the manifold conformance analysis

Threshold	Num DS	Total Wins			
		PCA	DAE	MOS	SMOTE
Comp					
< 0.176	21	18	20	20	1
≥ 0.176	12	7	4	7	5

on 20 of these datasets. This provides an accurate indication of when to apply manifold-based synthetic oversampling.

Finally, we note that there is also potential to use $m_{\mathcal{F}_i}(\cdot)$ to decide which manifold-based synthetic oversampling method to apply. Table 6 shows that using FACT and COMP for this task leads to the strongest correlation with the relative performance. Given the wide range of manifold learning methods available in the literature with their various biases and resulting complimentary strengths, it is very useful to have a method to select between them. This requires additional experimentation. Nonetheless, these results provide a good indication that $m_{\mathcal{F}_i}(\cdot)$ score can be used for this as well as generally deciding when to apply manifold based oversampling.

Table 6 Correlation between difference $AUC(DAE) - AUC(PCA)$ and $m_{\mathcal{F}}(\cdot)$

	COMP	FACT	MAP	BIC	ABIC	DC	$\lambda > \mu$	PA	OC	AF	PL	Dim
DIFF	−0.53	−0.60	0.18	0.15	0.05	−0.28	−0.24	−0.50	−0.40	−0.46	−0.47	0.18

8 Future work

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 aspect of 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 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; however, it is our expectation a model selection method designed specifically for synthetic oversampling would be ideal.

We are interested in studying the effectiveness of manifold learning techniques that rely on the local neighbourhood training points to produce their representation of the manifold. Our intuition suggests that the small training sets available in imbalanced domains will limit their effectiveness. 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 theories regarding how best to do this for the denoising autoencoder formalization, and continue to test this understanding and consider how to extend it to alternate methods.

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. Can it be applied in an organic way rather than breaking the task into a set of one versus one or one versus all problems, as is often done. Fecker et al. (2013) proposed the use of Gaussian mixture models for modelling and generating joint distribution of an imbalanced binary class problem. Their method involved modelling and generating from the joint distribution, and then apply applying an assignment rule to label the data for use in the induction of the classifier. Although their specific method was proposed for binary classification and is inappropriate for data that conforms to the manifold property, it suggests a potential means of extending our framework to imbalanced multi-class classification.

Deep learning algorithms have been extremely successful in a wide variety of domains, such as speech and object recognition, drug discovery and genomics. They has, indeed, been at the nexus of exciting developments in machine learning for the last few years. To be effective, however, deep learning algorithms require large datasets for training. This places a significant limit on where and when the power of deep learning can be leveraged. 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. Moreover, since many of the deep learning algorithms are based on artificial neural networks that are related to DAEs, we question if the synthetic oversampling process cannot be built into deep nets in a manner that enables bootstrapping based on examples that they themselves generated.

9 Conclusion

Our research into the imbalanced classification of gamma-ray spectra with the Radiation Protection Bureau at Health Canada led us to consider how to appropriately synthesize the minority class for this particular domain. This study led us to understand the negative impacts of conformance to the manifold property on the state-of-the-art in synthetic oversampling. We have shown that ignoring this leads to instances being generated in erroneous regions of the data space.

We proposed a general framework for manifold-based synthetic oversampling that first assesses the conformance to the manifold property in order to aid in selecting an appropriate method. The framework is then applied to model and generate additional training samples that spread along the embedded probability density; this leads to more realistic synthetic samples. 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. In addition, we demonstrated that our method is highly beneficial to challenging image classification tasks that commonly appear in the manifold learning and deep learning literature. Finally, we employed 16 benchmark datasets from the UCI repository to show, generally, that our framework leads to improved AUC results when the data conforms to the manifold property.

Acknowledgements Funding was provided by Canadian Network for Research and Innovation in Machining Technology, Natural Sciences and Engineering Research Council of Canada, Health Canada, Ontario Graduate Scholarship.

References

Akbani, R., Kwek, S., & Japkowicz, N. (2004). Applying support vector machines to imbalanced datasets. *Machine Learning: ECML*, *3201*, 39–50.

Alain, G., & Bengio, Y. (2014). What regularized auto-encoders learn from the data generating distribution. *The Journal of Machine Learning Research*, *15*(1), 3563–3593.

Alpaydin, E. (2014). *Introduction to machine learning* (3rd ed.). Cambridge, MA: MIT Press.

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

Batista, G. E., Bazzan, A. L. C., & Monard, M. C. (2003). Balancing training data for automated annotation of keywords: A case study. In *Brazilian workshop on bioinformatics* (pp. 10–18).

Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, *15*(6), 1373–1396.

Belkin, M., & Niyogi, P. (2004). Semi-supervised learning on Riemannian manifolds. *Machine Learning*, *56*, 209–239.

Bellinger, C. (2016). *Beyond the boundaries of SMOTE: A framework for manifold-based synthetic oversampling*. PhD thesis.

Bellinger, C., Drummond, C., & Japkowicz, N. (2016). Beyond the boundaries of SMOTE: A framework for manifold-based synthetic oversampling. In *European conference on machine learning* (pp. 1–16).

Bellinger, C., Japkowicz, N., & Drummond, C. (2015). Synthetic oversampling for advanced radioactive threat detection. In *International conference on machine learning and applications*. doi:10.1109/ICMLA.2015.58.

Blondel, M., Seki, K., & Uehara, K. (2011). Tackling class imbalance and data scarcity in literature-based gene function annotation. In *Proceedings of the 34th international ACM SIGIR conference on research and development in information—SIGIR'11* (pp. 1123–1124). New York, NY: ACM Press.

Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In *Pacific-Asia conference on knowledge discovery and data mining* (pp 475–482). Berlin: Springer.

Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate Behavioral Research*, *1*(2), 245–276.

Chapelle, O., Scholkopf, B., & Zien, A. (2006). *Semi-supervised learning*. Cambridge: MIT Press.

Chawla, N., Bowyer, K., Hall, L., & WP, K. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Chawla, N., Japkowicz, N., & Kolcz, A. (2003). Workshop learning from imbalanced data sets II. In *International conference on machine learning*.

Chawla, N. V., Japkowicz, N., & Drive, P. (2004). Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKD Explorations Newsletter*, *6*(1), 2000–2004.

Chawla, N., & Zhou, Z. H. (2009). Data mining when classes are imbalanced and errors have costs. In *Workshop, 13th Pacific-Asia conference on knowledge discovery and data mining*.

Courtney, M., & Ray, G. (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.

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

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78.

Drummond, C., & Holte, R. C. (2003). C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In *ICML workshop on learning from imbalanced datasets II*.

Fecker, D., Märgner, V., & Fingscheidt, T. (2013). Density-induced oversampling for highly imbalanced datasets. In P. R. Bingham & E. Y. Lam (Eds.), *SPIE. 8661, image processing: Machine vision applications VI* (Vol. 8661, pp. 86610P-1–86610P-11). Bellingham, WA: Society of Photo-Optical Instrumentation Engineers (SPIE).

Gao, M., Hong, X., Chen, S., & Harris, C. J. (2012). Probability density function estimation based oversampling for imbalanced two-class problems. In *The 2012 international joint conference on neural networks (IJCNN)* (pp. 1–8). IEEE.

Garrido, L. E., Abad, F. J., & Ponsoda, V. (2011). Performance of Velicer's minimum average partial factor retention method with categorical variables. *Educational and Psychological Measurement*, *71*, 551–570.

Gauld, D. B. (2008). Topological properties of manifolds. *The American Mathematical Monthly*, *81*(6), 633–636.

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

Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In D. S. Huang, X. P. Zhang, & G. B. Huang (Eds.), *Advances in intelligent computing*. *ICIC 2005*. Lecture Notes in Computer Science (Vol. 3644). Springer, Berlin, Heidelberg.

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)* (no. 3, pp. 1322–1328).

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284. doi:10.1109/TKDE.2008.239.

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

Humphreys, L. G., & Montanelli, R. G. J. (1975). An investigation of the parallel analysis criterion for determining the number of common factors. *Multivariate Behavioral Research*, *10*(2), 193–205.

Huo, X., Ni, X. S., & Smith, A. K. (2007). A Survey of manifold-based learning methods. In *International workshop on mining of enterprise data at recent advances in data mining of enterprise data: Algorithms and applications* (pp. 691–745). Singapore: World Scientific.

Japkowicz, N. (2000). Editor. In *AAAI'2000 workshop on learning from imbalanced data sets*.

Japkowicz, N. (2001). Supervised versus unsupervised binary-learning by feedforward neural networks. *Machine Learning*, *42*(1), 97–122. doi:10.1023/A:1007660820062.

Jo, T., & Japkowicz, N. (2004). Class imbalances versus small disjuncts. *ACM SIGKDD Explorations Newsletter: Special Issue on Learning from Imbalanced Datasets*, *6*(1), 40–49.

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

Kangas, L. J., Keller, P. E., Siciliano, E. R., Kouzes, R. T., & 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.

Krizhevsky, A. (2009). *Learning multiple layers of features from tiny images*. Master's thesis, Department of Computer Science, University of Toronto.

Kubat, M., Holte, R. C., & Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning, 30*(2), 195–215.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Liu, M., Wang, R., Huang, Z., Shan, S., & Chen, X. (2013). Partial least squares regression on grassmannian manifold for emotion recognition. In *Proceedings of the 15th ACM on international conference on multimodal interaction* (pp. 525–530). ACM.

Lui, Y. M., Beveridge, J. R., & Kirby, M. (2010). Action classification on product manifolds. In *2010 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 833–839). IEEE.

Ma, Y., & Fu, Y. (Eds.). (2011). *Manifold learning theory and applications*. CRC Press.

Nguwi, Y. Y., & Cho, S. Y. (2009). Support vector self-organizing learning for imbalanced medical data. In *2009 international joint conference on neural networks* (pp. 2250–2255). IEEE.

Olmos, P., Diaz, J., Perez, J., Gomez, P., Rodellar, V., Aguayo, P., et al. (1991). A new approach to automatic radiation spectrum analysis. *IEEE Transactions on Nuclear Science*, *38*(4), 971–975.

Raiche, G., Roipel, M., & Blais, J. G. (2006). Non graphical solutions for the Cattell's scree test. In *The international annual meeting of the psychometric*.

Revelle, W. (2013). *psych: Procedures for personality and psychological research*. Evanston, IL: Northwestern University. http://CRAN.R-project.org/package=psych Version = 1.3.2.

Revelle, W., & Rocklin, T. (1979). Very simple structure: An alternative procedure for estimating the optimal number of interpretable factors. *Multivariate Behavioral Research*, *4*(14), 403–414.

Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*(5500), 2323–2326.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 1, pp. 318–362). MA: The MIT Press.

Ruscio, J., & 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.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*, 461–464.

Silva, V. D., & Tenenbaum, J. B. (2003). Global versus local methods in nonlinear dimensionality reduction. In *Advances in neural information processing systems* (Vol. 15, pp. 721–728). Cambridge, MA: The MIT Press.

Slama, R., Wannous, H., Daoudi, M., & Srivastava, A. (2015). Accurate 3D action recognition using learning on the Grassmann manifold. *Pattern Recognition*, *48*(2), 556–567.

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

Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science (New York, NY)*, *290*(5500), 2319–23.

Tuzel, O., Porikli, F., & Mee, P. (2007a). Human detection via classification on Riemannian manifolds. In *2017 IEEE conference on computer vision and pattern recognition*, Minneapolis, MN, 2007 (pp. 1–8).

Tuzel, O., Porikli, F., & Meer, P. (2007b). Human detection via classification on riemannian manifolds. In *IEEE conference on computer vision and pattern recognition, 2007. CVPR'07* (pp. 1–8). IEEE.

Tuzel, O., Porikli, F., & Meer, P. (2008). Pedestrian detection via classification on riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(10), 1713–1727.

Van Hulse, J., Khoshgoftaar, T. M., & Napolitano, A. (2007). Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on machine learning* (pp. 935–942).

Velicer, W. (1976). Determining the number of components from the matrix of partial correlations. *Psychometrika*, *41*(3), 321–327.

Vincent, P. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising. *Criterion*, *11*, 3371–3408.

Wallace, B. C., Small, K., Brodley, C. E., & Trikalinos, T. A. (2011). Class imbalance, redux. In *2011 IEEE 11th international conference on data mining* (pp. 754–763). IEEE.

Wang, S., Minku, L. L., Chawla, N., & Yao, X. (2017). Workshop on learning in the presence of class imbalance and concept drift. In *IJCAI 2017 workshop*.

Wei, J., Peng, H., Lin, Y.-S., Huang, Z.-M., & Wang, J.-B. (2008). Adaptive neighborhood selection for manifold learning. In *2008 international conference on machine learning and cybernetics* (Vol. 1, pp. 380–384).

Weinberger, K. Q., Sha, F., & Saul, L. K. (2004a). Learning a kernel matrix for nonlinear dimensionality reduction. In *International conference on machine learning* (pp. 106–113).

Weinberger, K. Q., Sha, F., & Saul, L. K. (2004b). Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the twenty-first international conference on machine learning* (p. 106). ACM.

Weiss, G. M. (2004). Mining with rarity: A unifying framework. *SIGKDD Explorations Newsletter*, *6*(1), 7–19. doi:10.1145/1007730.1007734.

Xue, H. U. I., & Chen, S. C. (2007). Alternative robust local embedding. In *2007 international conference on wavelet analysis and pattern recognition* (pp. 591–596).

Yang, Q., Wu, X., Elkan, C., Gehrke, J., Han, J., Heckerman, D., et al. (2006). 10 challenging problems in data mining research. *International Journal of Information Technology and Decision Making*, *5*(4), 597–604.

Yoshida, E., Shizuma, K., Endo, S., & 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.

Zhang, D., & 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* (pp. 266–273).

Zhu, M., & Ghodsi, A. (2006). Automatic dimensionality selection from the scree plot via the use of profile likelihood. *Computational Statistics and Data Analysis*, *51*(2), 918–930.