Knowl Inf Syst
DOI 10.1007/s10115-010-0310-3

REGULAR PAPER

Improving SVM classification on imbalanced time series data sets with ghost points

Suzan Köknar-Tezel · Longin Jan Latecki

Received: 7 March 2010 / Revised: 7 April 2010 / Accepted: 15 May 2010

Abstract Imbalanced data sets present a particular challenge to the data mining community. Often, it is the rare event that is of interest and the cost of misclassifying the rare event is higher than misclassifying the usual event. When the data is highly skewed toward the usual, it can be very difficult for a learning system to accurately detect the rare event. There have been many approaches in recent years for handling imbalanced data sets, from under-sampling the majority class to adding synthetic points to the minority class in feature space. However, distances between time series are known to be non-Euclidean and non-metric, since comparing time series requires warping in time. This fact makes it impossible to apply standard methods like SMOTE to insert synthetic data points in feature spaces. We present an innovative approach that augments the minority class by adding synthetic points in *distance spaces*. We then use Support Vector Machines for classification. Our experimental results on standard time series show that our synthetic points significantly improve the classification rate of the rare events, and in most cases also improves the overall accuracy of SVMs. We also show how adding our synthetic points can aid in the visualization of time series data sets.

Keywords Imbalanced data sets · Support Vector Machines · Time series

1 Introduction

Most traditional learning systems assume that the class distribution in data sets is balanced, an assumption that is often violated. There are many real-world applications where the data sets are highly imbalanced, such as oil spill detection from satellite images [16], credit card fraud detection [7], medical diagnostics [21], and predicting telecommunication equipment failure [31]. In these data sets, there are many examples of the "normal" (the majority/negative

S. Köknar-Tezel (✉) · L. J. Latecki
Department of Computer and Information Sciences,
Temple University, Philadelphia, PA, USA
e-mail: tezel@temple.edu

Published online: 16 June 2010

class), and very few examples of the "abnormal" (the minority/positive class). But often it is the rare occurrence, the "abnormal", which is the interesting or important occurrence, e.g. an oil spill. In data mining, the rare occurrence is usually much more difficult to identify since there are so few examples and most traditional learning systems are designed to work on balanced data. These learning systems are biased toward the majority class, focus on improving overall performance, and usually perform poorly on the minority class. If a data set has say 999 examples of the normal event and only one example of the abnormal event, a learning system that predicts all examples as "normal" will be 99.9% accurate, but misclassify the very important abnormal example.

Mining imbalanced data sets has been the focus of much research recently [4,10,30], and one important direction is sampling strategies. Sampling methods may include removing majority class data points (under-sampling) or inserting minority class data points (over-sampling) in order to improve accuracy. Two well-known techniques for increasing the number of minority examples are random resampling and SMOTE (Synthetic Minority Over-sampling TEchnique) [8]. In random resampling, minority class examples are randomly replicated, but this can lead to overfitting. The SMOTE algorithm inserts synthetic data into the original data set to increase the number of minority class examples. The synthetic points are generated from existing minority class examples by taking the difference between the corresponding feature values of a minority class example x and one of its nearest neighbors in the minority class, multiplying each feature difference by a random number between 0 and 1, and then adding these amounts to the feature vector of x.

SMOTE and its variations, for example [9,3,13], have shown that they can improve overall classification accuracy and also improve the learning of the rare event. But SMOTE and its variations work only in feature space, i.e., each example is represented as a point in n-dimensional space where n is the number of features of each example. However, for some fields such as bioinformatics, image analysis, and cognitive psychology, often the feature vectors are not available. Instead, in these domains, the data may be represented as a matrix of pairwise comparisons where typically each element of the matrix is the distance (similarity or dissimilarity) between the corresponding original data points. This matrix represents the *distance space* of the data. Often, this distance space is non-metric because the distance function used to calculate the similarities or dissimilarities between the pairs of data points does not satisfy the mathematical requirements of a metric function. For example, the distances between time series are often non-metric due to warping. When only pairwise scores are available, the feature space based approaches to adding synthetic points cannot be used. In our experiments, we do not compare ghost points with SMOTE or random resampling because SMOTE and random resampling do not work in distance spaces, while the distinct advantage of the proposed approach is that it can be used in distance spaces. Our approach to balancing the data sets is to use supervised learning to increase the size of the minority class by inserting synthetic points directly into the distance space. Our synthetic points do not have any coordinates, i.e., they are not points in any vector space, which is why we call our synthetic points *ghost points*. But our ghost points *are* points in distance space.

To show the flexibility of our approach, we inserted ghost points into the distance spaces induced by two different distance measures, Dynamic Time Warping (DTW) [5,23] and Optimal Subsequence Bijection (OSB) [17]. For a nice overview of elastic sequence matching algorithms, see [12]. The DTW distance between two sequences is the sum of distances of their corresponding elements. Dynamic programming is used to find corresponding elements so that this distance is minimal. The DTW distance has been shown to be superior to the Euclidean distance in many cases [1,36]. However, DTW is particularly sensitive to outliers, since it is not able to skip any elements of the sequences. In DTW, each element of

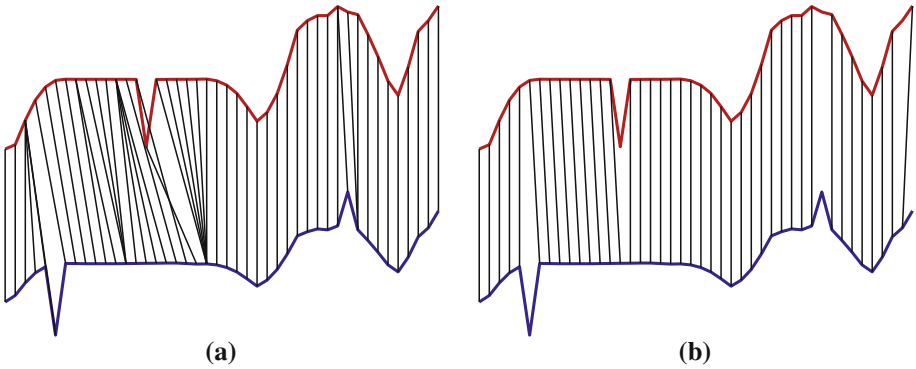


Fig. 1 The *top* and *bottom* sequences represent parts of contours of two different but very similar bone shapes. The correspondence obtained by DTW is shown in (**a**). The correspondence obtained by OSB is shown in (**b**)

the query sequence must correspond to some element of the target sequence and vice versa (see Fig. 1a). Thus, the optimal correspondence computed by DTW is a relation on the set of indices of both sequences, i.e., a one-to-many and many-to-one mapping. The fact that outlier elements must participate in the correspondence optimized by DTW often leads to an incorrect correspondence of other sequence elements. OSB computes the distance value between two sequences based directly on the distances of corresponding elements just as DTW does, but unlike DTW, OSB can skip outlier elements of the query and target sequences when computing the correspondence (see Fig. 1b). This makes the performance of OSB robust in the presence of outliers. Moreover, OSB defines a bijection on the matched subsequences, which means that we have a one-to-one correspondence of the matched elements.

We choose Support Vector Machines (SVMs) to perform the classification because they are a fundamental machine learning tool and they have a strong theoretical foundation [27], though ghost points can also be used with newer methods like Lotka–Volterra derived models [14]. SVMs have been very successful in pattern recognition and data mining applications on balanced data sets. But when data sets are unbalanced, the SVM's accuracy on the minority/positive examples is poor. This is because the class-boundary learned by the SVM is skewed toward the majority/negative class [34]. This may lead to many positive examples being classified as negative (false negatives), which in some situations can be very costly (e.g. missing an oil spill, missing a cancer diagnosis). There are cost-sensitive SVMs that assign different costs to different classification errors and the SVM attempts to minimize misclassification costs instead of maximizing accuracy. Often though, in many real-world situations, the misclassification costs are unknown. Also, how to assign the costs is still an active research area and has not been solved. For example, [38] discusses two major approaches to converting traditional classifiers into cost-sensitive classifiers, and [28] combines modifying the classifier with changing the data distribution. Using ghost points eliminates the need for cost-sensitive classifiers and our experimental results (see Sect. 4) show that inserting ghost points in both DTW distance spaces and OSB distance spaces can significantly increase the SVM's ability to learn the rare events. Furthermore, in most cases, the addition of ghost points increases the SVM's overall classification accuracy.

In Sect. 2, we introduce the definition of ghost points. In Sect. 3, we demonstrate how ghost points can be used to visualize data, particularly minority classes. We discuss evaluating performance on imbalanced data sets and our experimental results in Sect. 4. In Sect. 5, we summarize and discuss our future work.

2 Definition of ghost points

In many applications, only distance (or equivalently similarity) information is available, in which case operations in vector space cannot generate synthetic points. This is the case when the data points do not have any coordinates, or the data points have coordinates but the Euclidean distance does not reflect their structure. Consequently, a distance measure is used that is not equivalent to the Euclidean distance, e.g., [5,17]. For this type of data, researchers usually utilize embeddings to low-dimensional Euclidean spaces. However, embedding implies distance distortion. It is known that not every four point metric space can be isometrically embedded into an Euclidean space $\mathbb{R}^k$, e.g., see [20].

Definition 2.1 A *metric* on a set X is a distance function $\rho : X \times X \to \mathbb{R}$, such that the following axioms hold:

1. $\rho(x, y) \geq 0$ (non-negativity)
2. $\rho(x, y) = \rho(y, x)$ (symmetry)
3. $\rho(x, y) = 0 \Leftrightarrow x = y$ (positive definiteness)
4. $\rho(x, y) + \rho(y, z) \geq \rho(x, z)$ (triangle inequality)

for any $x, y, z \in X$.

Definition 2.2 A *metric space* is an ordered pair (X, ρ), where X is a set of points, and ρ is metric on X, that is, a distance function $\rho : X \times X \to \mathbb{R}$.

Definition 2.3 Let Y and Z be two metric spaces. We say that a mapping f of the space Y into Z is an *isometric embedding* if $dist_Z(f(y_1), f(y_2)) = dist_Y(y_1, y_2)$.

A simple example where distances are not preserved when mapping a four point metric space to $\mathbb{R}^k$ is presented in [11]. Given the metric space (X, ρ) defined in Fig. 2a, assume there exists a mapping $f : X = \{a, b, c, d\} \to \mathbb{R}^k$ for some k where f preserves the distances. The triangle inequality holds for the elements a, b, and d; in fact $\rho(b, d) = \rho(b, a) + \rho(a, d)$ and because of the equality, the mapped points $f(b)$, $f(a)$, and $f(d)$ are collinear in the space $\mathbb{R}^k$. This also holds for points a, c, and d, i.e., they are collinear in $\mathbb{R}^k$ (Fig. 2b). Since both lines have two points in common, they must be the same line (Fig. 2c). But then $f(b) = f(c)$ contradicting the fact that the original distance between b and c is 2. Therefore, the assumption that f preserves the distances is false.

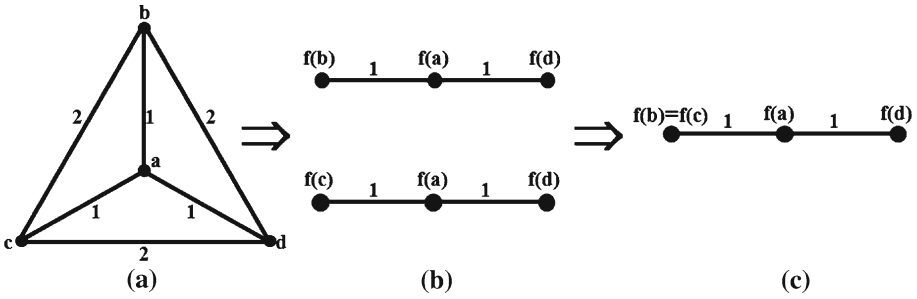


Fig. 2 (**a**) Example of a 4-point metric space that cannot be embedded into a Euclidean space. (**b**) The points b, a, and d are collinear after embedding, and so are the points c, a, and d. Thus, points b and c are the same point after embedding as shown in (**c**)

Definition 2.4 In this paper, a *distance space* is an ordered pair (X, ρ), where X is a set of points and $\rho : X \times X \to \mathbb{R}$ is a distance function that satisfies the first two axioms and the $\Leftarrow$ direction of axiom 3 from Definition 2.1.

Clearly, we would like ρ to be as close as possible to a metric, but this is not always possible, e.g., there are clear arguments from human visual perception that the distances induced by human judgments are often non-metric [19].

The key observation of the proposed approach is that although not every four point metric space can be embedded into a Euclidean space, every three point metric space can be isometrically embedded into the plane $\mathbb{R}^2$. Let (Δ, ρ), where $\Delta = \{x, a, b\} \subseteq X$, be a metric space with three distinct points. Then, it is easy to map Δ to the vertices of a triangle on the plane. For example, we can construct an isometric embedding $h : \Delta \to \mathbb{R}^2$ by setting $h(a) = (0, 0)$ and $h(b) = (\rho(a, b), 0)$. Then, $h(x)$ is uniquely defined as a point with non-negative coordinates such that its Euclidean distance to $h(a)$ is $\rho(x, a)$ and its Euclidean distance to $h(b)$ is $\rho(x, b)$. $h : \Delta \to \mathbb{R}^2$ is an isometric embedding, since for any two points $y, z \in \Delta$, $\rho(y, z)^2 = ||y - z||^2$, where $|| \cdot ||$ is the standard L_2 norm that induces the Euclidean distance on the plane. We stress that this construction does not require that (X, ρ) be a metric space, but it does require that the three point space (Δ, ρ) be a metric space. Below we will generalize this construction to the case when Δ is not a metric space.

Definition 2.5 Given any two points a, b in a distance space X, we define a *ghost point* e induced by a and b using the construction $e = \mu(a, b) = h^{-1}(\frac{1}{2}(h(a) + h(b))$. For every $x \in X$, the distance from x to e, $\rho(x, \mu(a, b))$, is computed as follows:

1. If the three point subspace $\Delta = \{x, a, b\}$ is a metric, then use Eq. 1 below.
2. If $\rho(a, b) > \rho(x, a) + \rho(x, b)$, then use Eq. 2 below.
3. (a) If $\rho(x, a) > \rho(x, b) + \rho(a, b)$, then use Eq. 3 below or
   (b) If $\rho(x, b) > \rho(x, a) + \rho(a, b)$, then use Eq. 4 below.

Cases 2 and 3 in this definition apply when Δ is not a metric space.

Let $\mu(a, b)$ denote the mean of two points a, b. If $a, b \in \mathbb{R}$, then we have the usual formula $\mu(a, b) = \frac{1}{2}(a + b)$ (see Fig. 3a, where red points are original data, the green point e is the ghost point and $e = \mu(a, b)$).

Our first key contribution is the definition of $\mu(a, b)$ for any two points a, b in a distance space X. To define $\mu(a, b)$, we need to specify $\rho(x, \mu(a, b))$ for every $x \in X$. There are three cases depending on whether the three point subspace $\Delta = \{x, a, b\} \subseteq X$ is a metric or not.

Case 2.1 Type1: $\Delta = \{x, a, b\} \subseteq X$ is a metric subspace

We first isometrically embed Δ into the plane $\mathbb{R}^2$ by h. We define $\mu(a, b) = h^{-1}(\frac{1}{2}(h(a) + h(b))$. Since $h(\Delta)$ defines vertices of a triangle on the plane, we can easily derive that

$$||h(x) - \frac{h(a) + h(b)}{2}||^2 = \frac{||h(x) - h(a)||^2}{2} + \frac{||h(x) - h(b)||^2}{2} - \frac{||h(a) - h(b)||^2}{4}$$

Since h is an isometry and $\mu(a, b) = h^{-1}(\frac{1}{2}(h(a) + h(b))$, we obtain (see Fig. 3a)

$$\rho(x, \mu(a, b))^2 = \frac{1}{2}\rho(x, a)^2 + \frac{1}{2}\rho(x, b)^2 - \frac{1}{4}\rho(a, b)^2 \quad (1)$$

Consequently, Eq. 1 defines the distance of every point $x \in X$ to the new point $\mu(a, b)$, which we call the mean of a and b. By computing the distances of $\mu(a, b)$ to all points

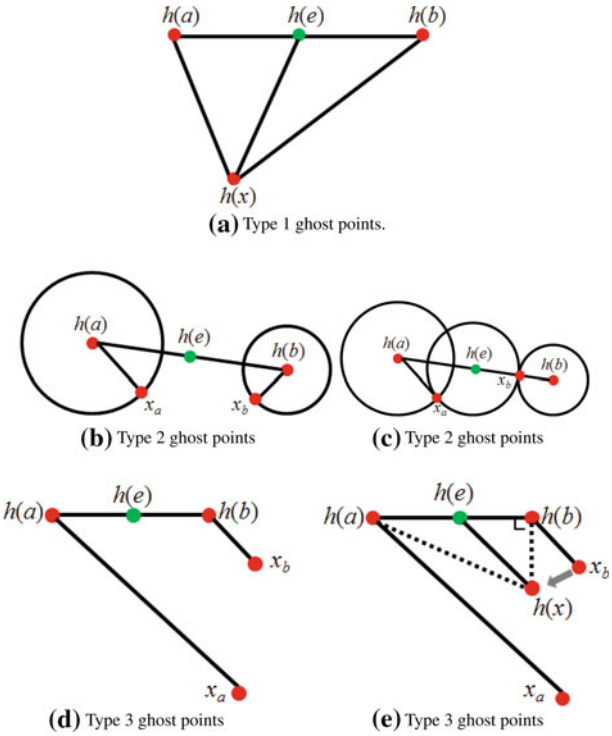


Fig. 3 **a** The construction of $\rho(x, e)$ for $e = \mu(a, b)$ for a triple of points that satisfy the triangle inequality. **b** Triple of points that cannot construct a triangle. The way to calculate $\rho(x, e)$ for 3b is shown in (**c**). Another way in which the triangle inequality is violated is shown in (**d**) and the approach to calculating $\rho(x, e)$ is shown in (**e**)

in X, we define a new point $\mu(a, b)$, and the augmented set $X' = X \cup \{\mu(a, b)\}$ is also a distance space. We stress that to add a new point $\mu(a, b)$ to X we do not need to compute the embedding h. We use h only to derive Eq. 1. Moreover, since the embedding h is an isometry, Eq. 1 defines locally correct distances from $\mu(a, b)$ to all points in X. Since we can compute the correct distances without explicitly computing the mapping h, this is similar to the kernel trick [2].

Case 2.2 Type 2: $\Delta = \{x, a, b\} \subseteq X$ is not a metric subspace and $\rho(a, b) > \rho(x, a) + \rho(x, b)$

In Eq. 1, we assume that the three point space (Δ, ρ) is a metric space. Thus, we assume that the local structure of any distance space X can be locally approximated by the metric space, which is also the assumption for embedding approaches [22,24]. However, for some point triples $\Delta = \{x, a, b\} \subseteq X$, (Δ, ρ) is not a metric space, which may lead to a negative distance in Eq. 1. This is the case if $\rho(a, b) > \rho(x, a) + \rho(x, b)$. Then a triangle with vertices $h(a)$, $h(b)$, and $h(x)$ cannot be constructed on the plane, as illustrated in Fig. 3b. Since a single point $h(x)$ on the plane does not exist, we map $h(x)$ to two different points denoted x_a and x_b such that $\rho(x, a) = ||h(a) - x_a||$ and $\rho(x, b) = ||h(b) - x_b||$. Without loss of generality we assume that $\rho(x, a) > \rho(x, b)$. Then it is possible to position points x_a and x_b on the plane such that (see Fig. 3c): $\rho(x, a) = ||h(a) - x_a||$, $\rho(x, b) = ||h(b) - x_b||$, and $||h(\mu(a, b)) - x_a|| = ||h(\mu(a, b)) - x_b||$.

Thus, both points x_a and x_b are the same distance away from $h(\mu(a,b))$, and this distance is equal to $\frac{1}{2}||h(a)-h(b)||-||x_b-b||$. Therefore, we define $h(x)=\{x_a,x_b\}$ and

$$\rho(x,\mu(a,b))=\frac{1}{2}\rho(a,b)-\rho(x,b) \tag{2}$$

Formally, h maps x to a single point in a quotient space $\mathbb{R}^2/\{x_a,x_b\}$, and h remains an isometric embedding but to the quotient space.

Case 2.3 Type 3: $\Delta=\{x,a,b\}\subseteq X$ is not a metric subspace and either $\rho(x,a)>\rho(x,b)+\rho(a,b)$ or $\rho(x,b)>\rho(x,a)+\rho(a,b)$

In this case, as in Case 2.2, (Δ,ρ) is not a metric space and again may lead to a negative distance in Eq. 1. This occurs if either $\rho(x,a)>\rho(x,b)+\rho(a,b)$ or $\rho(x,b)>\rho(x,a)+\rho(a,b)$. Then a triangle with vertices $h(a),h(b),h(x)$ cannot be constructed on the plane, as illustrated in Fig. 3d. Since a single point $h(x)$ on the plane does not exist, we again map $h(x)$ to two different points denoted x_a and x_b such that $\rho(x,a)=||h(a)-x_a||$ and $\rho(x,b)=||h(b)-x_b||$.

Without loss of generality we assume that $\rho(x,a)>\rho(x,b)+\rho(a,b)$. In this case, we first position point x_b on the plane so that the angle $h(a)h(b)x_b$ is straight without changing the distance from $h(b)$ to x_b (see Fig. 3e). Then, we use the triangle $h(a)h(b)x_b$ to define the ghost point. When doing so we ignore the distance $\rho(x,a)$ in this construction or equivalently, only consider the assignment $h(x)=x_b$. Unlike Case 2.2, it is impossible to make the assignments $h(x)=x_a$ and $h(x)=x_b$ consistent, hence we need to ignore one of them. Since $\rho(x,b)$ is significantly smaller than $\rho(x,a)$, and small distances are less likely to be the result of noise, we rely only on $h(x)=x_b$. We can then use the right triangle $h(a)h(b)x_b$ to define

$$\rho(x,\mu(a,b))^2=\rho(x,b)^2+\frac{1}{4}\rho(a,b)^2. \tag{3}$$

Similarly, if $\rho(x,b)>\rho(x,a)+\rho(a,b)$, we define

$$\rho(x,\mu(a,b))^2=\rho(x,a)^2+\frac{1}{4}\rho(a,b)^2. \tag{4}$$

The so-defined distances to ghost points are guaranteed to be non-negative and symmetric by their construction. Hence the space augmented by ghost points remains a distance space. However, it may happen that two different points have distance zero, and this is possible even if X is a metric space. For example, assume that X is a sphere of radius 1 and that points a and b are on the north and south poles (see Fig. 4). For any point $x\in X$ on the equatorial line the distance between $\mu(a,b)$ and x becomes $\rho(x,\mu(a,b))^2=0.5(\pi/2)^2+0.5(\pi/2)^2-0.25\pi^2=0$. Therefore, every point on the equatorial line has a distance of 0 to the ghost point $\mu(a,b)$. This example also shows that adding ghost points to a metric space may lead to a non-metric space. We stress however that the intended application of the proposed method is to densify distance spaces that are non-metric, since such spaces are common in many cognitively motivated tasks such as distances between images, shapes, text documents, and so on. We also stress that though global metricity is not necessary, local metricity is preferred. If the triple of points a, b, and x is close to a metric, then the embedding of the three points is uniquely defined and Eq. 1 can be used to calculate the distance between x and the ghost point.

If the space X is finite, i.e., $X=\{x_1,\ldots,x_n\}$, then the distance function $\rho: X\times X\to\mathbb{R}_{\geq 0}$ is represented by a square matrix $M_\rho(X)$. Each row of the square distance matrix $M_\rho(X)$ is the distance of one data point x to all data points in the data set, i.e., for all $y\in X$,

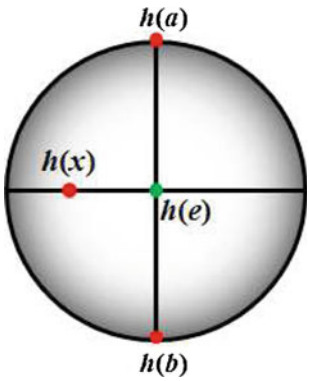


Fig. 4 Example of a unit sphere where $\rho(h(e), h(x)) = 0$

$M_\rho(x, y) = \rho(x, y)$. The matrix for $X \cup \{\mu(a, b)\}$ is obtained by simply adding one row and one column to $M_\rho(X)$, with each entry computed using Eqs. 1, 2, 3, or 4.

Thus, the proposed approach can be applied to metric and non-metric distance spaces, and our construction guarantees that the distances to all ghost points are non-negative and symmetric. In Sect. 4.3, we show the results of experiments that count the number of Type 1, Type 2, and Type3 computations performed on eighteen data sets using distance spaces induced by two different distance functions, OSB and DTW.

3 Visualizing data

High-dimensional data, such as time series, are often hard to visualize, though visualization can help in the analysis of trends, periodicity, motifs, and the like. When the actual time series sequences are available, line graphs can be a very effective tool to visualize and analyze time series. One of the earliest known time series plot is of planetary orbits from a tenth century monastery [25]. There have been some advances in the visualization of time series (for example, [29]), but the line graph is still the most prevalent. But if instead of sequences, the data is represented as a distance space (pair-wise distances between each time series), then the visualization becomes much more difficult as line graphs are not sufficient. In addition, even when plotted in a graph, if the data set is imbalanced, the minority class is often undiscernible among all the points of the majority class.

Adding ghost points to the minority class before plotting can change the structure of the underlying points so that minority class clusters become visible. For example, Fig. 5 shows the Wafer training set before and after adding ghost points to the distance matrix induced by Optimal Subsequence Bijection (OSB). The training set has 903 samples of the majority class and 97 samples of the minority class for a total of 1000 samples. To create Fig. 5a, we first take the original 1000×1000 distance matrix and use principal component analysis (PCA) to reduce the dimensionality to two dimensions. For Fig. 5b we add 9 ghost points per minority sample to the distance matrix (to create a 1873×1873 matrix) and again run PCA. The majority class is plotted as light blue circles, the minority class as black squares, and the ghost points as green squares. In Fig. 5a, without ghost points, it is impossible to distinguish the minority class from the majority class since the minority class forms no cluster and many of the minority class points overlap the majority class clusters. In Fig. 5b, after ghost points are added to the training set, the underlying shape of the data changes to form five discernable

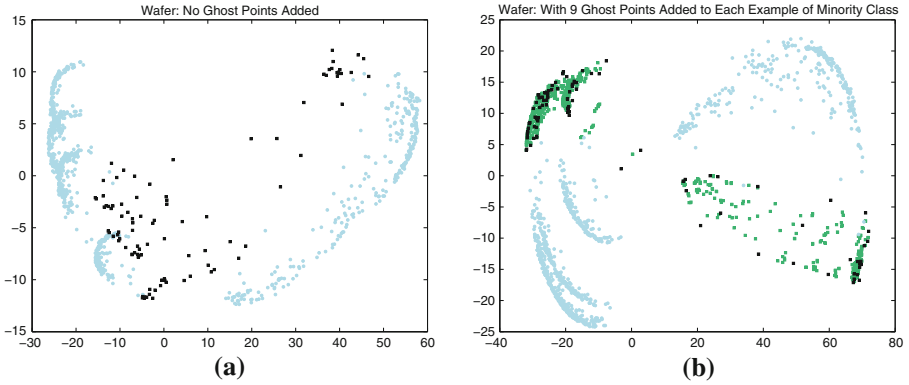


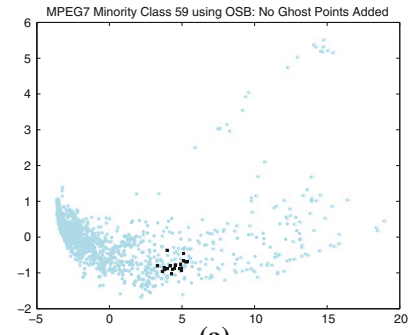
Fig. 5 After using PCA on the distance matrix to reduce the dimensionality from 1000 to 2, (**a**) the 1000 examples are plotted (the majority class as *light blue circles* and the minority class as *black squares*). **b** is the same data set with 9 ghost points (*green squares*) added per minority example. In (**a**), it is impossible to distinguish the minority class from the majority class as the minority class has no structure. However, in (**b**) there are 5 distinct clusters, 2 of which belong to the minority class (Best viewed in color)

clusters. It is clear that two of the clusters belong to the minority class (the upper-left cluster and the lower-right cluster).

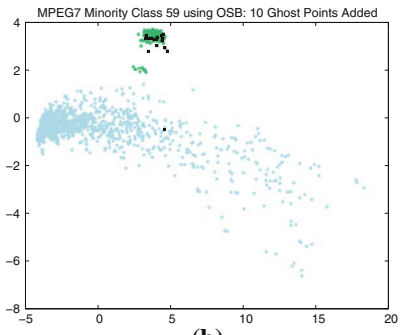
The next four examples are from the MPEG-7 image data set (see Sect. 4 for a description of the data set). In Fig. 6, the first two rows show the MPEG-7 data set before and after adding ghost points to the distance matrix induced by OSB for two different minority classes, *rat* and *teddy*; the second two rows show the MPEG-7 data set before and after adding ghost points to the distance matrix induced by DTW for the same two minority classes, *rat* and *teddy*. We divide the data set into 1380 samples of a majority class (69 classes of 20 images each, collapsed into a single class) and 20 samples of a minority class for a total of 1400 samples. As before, for Fig. 6a, c, e, and g, we take the original 1400×1400 distance matrix and use PCA to reduce the dimensionality to two and then plot the 1400 points. For Fig. 6b, d, f and h we add 10 ghost points for each of the 20 minority class samples to the distance matrix (creating a 1600×1600 matrix) and again run PCA. Again, a majority class is plotted as light blue circles, a minority class as black squares, and the ghost points as green squares. In Fig. 6a, c, e and g without ghost points, the minority class points overlap the majority class points and cannot be differentiated from the majority class points. In Fig. 6b, d, f and h, after ghost points are added to the data set, the underlying shape of the minority class changes to form distinct and visible clusters, with very few points, if any, overlapping the majority class points.

4 Experimental evaluation

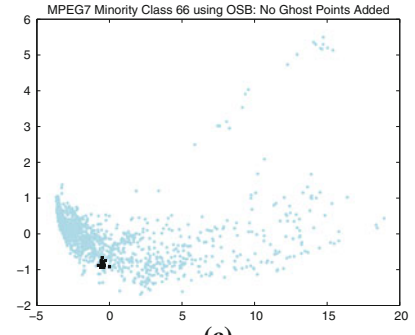
In many real-world situations, the minority class, the class with the fewest examples, is by far the most important class. Take for example the Mammography data set [33], which consists of non-calcification (non-cancerous) and calcification (cancerous) examples. The data set has 11183 examples of which only 260 (2.32%) are examples of cancer. A trivial classifier that classifies all examples as non-cancerous will achieve an accuracy of 97.68%, though its error rate for the minority class is 100%. For this data set, there are also uneven costs associated with misclassifying a normal example and misclassifying a cancerous example. If


MPEG7 Minority Class 59 using OSB: No Ghost Points Added

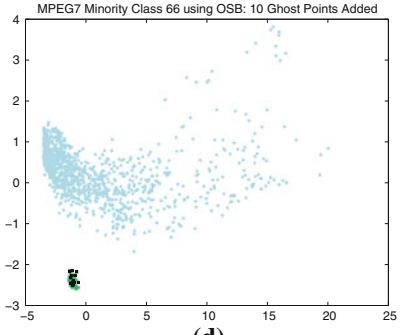
(a)


MPEG7 Minority Class 59 using OSB: 10 Ghost Points Added

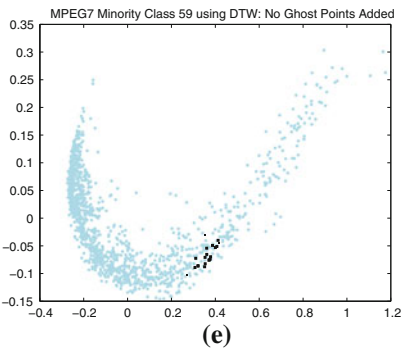
(b)


MPEG7 Minority Class 66 using OSB: No Ghost Points Added

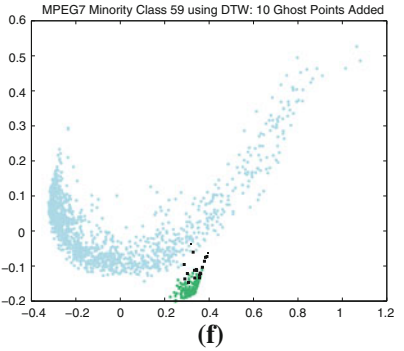
(c)


MPEG7 Minority Class 66 using OSB: 10 Ghost Points Added

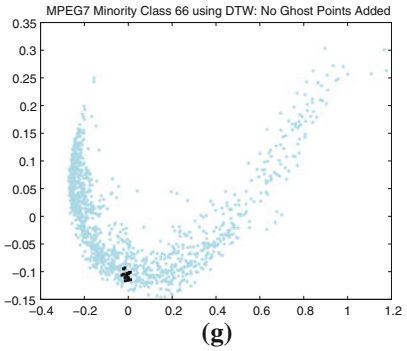
(d)


MPEG7 Minority Class 59 using DTW: No Ghost Points Added

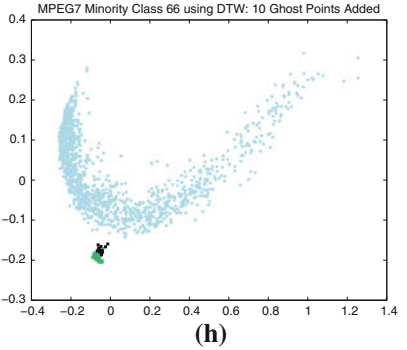
(e)


MPEG7 Minority Class 59 using DTW: 10 Ghost Points Added

(f)


MPEG7 Minority Class 66 using DTW: No Ghost Points Added

(g)


MPEG7 Minority Class 66 using DTW: 10 Ghost Points Added

(h)

◀**Fig. 6** The MPEG-7 data set. The first two rows use the distance matrix induced by OSB, the second two rows by DTW; rows 1 and 3 show the minority class rat and rows 2 and 4 show the minority class teddy. Column 1 shows the data sets without ghost points; column 2 after adding ghost points. After using PCA on the distance matrix to reduce the dimensionality from 1400 to 2, the 1400 examples are plotted ((**a**), (**c**), (**e**), and (**g**)); the majority class is plotted as *light blue circles* and the minority class as *black squares*. (**b**), (**d**), (**f**), and (**h**) are the same corresponding data sets with 10 ghost points (*green squares*) added per minority example. In (**a**), (**c**), (**e**), and (**g**), it is impossible to distinguish the minority class from the majority class as the minority class completely overlaps the majority class. However, in (**b**), (**d**), (**f**), and (**h**) the minority class now forms a distinct cluster (Best viewed in color)

a healthy patient is incorrectly diagnosed with having breast cancer, there is a cost associated with this error (fear, unnecessary tests) but eventually the misdiagnosis will be found. On the other hand, if a patient who does have breast cancer is incorrectly diagnosed as being healthy, then the cost could be her life since she will not get appropriate treatment. When the performance on the minority class is as important or more important than overall accuracy, other performance measures must be used. A common measure is F_β*-measure* [26] which is defined below in Sect. 4.1.

Unlike other techniques that add synthetic points, ghost points have the advantage that they can be added in distance space. To show that they will work with different distance measures, we use both DTW and OSB as distance measures on the UCR Time Series data sets [15] and on the MPEG-7 Core Experiment CE-Shape-1 data set [18] for our experiments.

The UCR Time Series data repository has available 20 data sets from various domains. The time series lengths range from 60 (Synthetic Control) to 637 (Lightning-2) and the number of classes in a data set ranges from 2 to 50. Each data set is divided into a fixed training set and testing set. The number of examples in a training set ranges from 24 (FaceFour) to 1000 (Wafer), and the number of testing examples ranges from 28 (Coffee) to 6174 (Wafer). In our experiments, we use seventeen of the data sets and their characteristics are described in Table 1.

MPEG-7 is a standard data set and is widely used to test shape classification and retrieval methods. It contains 1400 binary images consisting of 70 object classes (e.g. "Rats") and within each class there are 20 shapes, for a total of 1400 shapes. For our experiments, each shape is represented with 100 equidistant sample points on the contour, and these points are converted into sequences by calculating the curvature of each point with respect to its five neighbors on each side. This yields 1400 sequences of real numbers, each of length 100. This particular transformation makes the sequence representation invariant to rotation and scale changes. In other words, the shape of a cell phone with its antenna pointing up can still match with the same cell phone shape scaled and rotated so that the phone is now smaller and its antenna is pointing down.

4.1 Evaluating performance

Most studies on the class imbalance problem concentrate on two-class problems since multi-class data sets can easily be reduced to two classes (see Sect. 4.2). In an imbalanced data set, one class, the *majority* class or the *negative* class, has many examples, while the other class, the *minority* class or *positive* class, has few examples. These imbalances in real-world data sets can be 2:1, 1000:1, or even 10000:1. When a data set is imbalanced, the usual forms of evaluating performance do not work. For classification, generally the overall accuracy (the fraction of examples that are correctly classified) or the error rate ($1 - accuracy$) is reported, but this does not have much value if the interest lies in the minority class. It has been empirically shown that accuracy can lead to poor performance for the minority class [32]. Another

Table 1 The characteristics of the 17 UCR data sets used in our experiments

Data set name	Number of classes	Number of classes used as minority class	Training set			Testing set		
			Total number of examples	Number of minority examples	Number of majority examples	Total number of examples	Number of minority examples	Number of majority examples
SyntheticControl	6	6	300	50	250	300	50	250
CBF	3	2	30	8–10	20–22	900	300–302	598–600
FaceAll	14	14	560	40	520	1690	8–287	1403–1682
OSULeaf	6	6	200	15–53	147–185	242	23–55	187–219
SwedishLeaf	15	15	500	26–42	458–474	625	33–49	576–592
50Words	50	49	450	2–52	398–448	455	1–57	398–454
Trace	4	4	100	21–31	69–79	100	19–29	71–81
TwoPatterns	4	4	1000	237–271	729–763	4000	959–1035	2965–3041
Wafer	2	1	1000	97	903	6174	665	5509
FaceFour	4	4	24	3–8	16–21	88	14–26	62–74
Lightning2	2	1	60	20	40	61	28	33
Lightning7	7	7	70	5–19	51–65	73	6–19	54–67
ECG	2	1	100	31	69	100	36	64
Adiac	37	37	390	4–15	375–386	391	6–16	375–385
Fish	7	7	175	21–28	147–154	175	22–29	146–153
Beef	5	5	30	6	24	30	6	24
OliveOil	4	3	30	4–8	22–26	30	4–9	21–26

Table 2 Confusion matrix

	Predicted positive	Predicted negative
Actual positive	**TP**	**FN**
Actual negative	**FP**	**TN**

problem with using accuracy as the performance metric is that different classification errors are given the same importance, whereas in actuality their costs might differ significantly. One solution commonly used is to have a weighted loss function with higher loss for the minority class [6], but it requires knowing the loss weights, which is often impossible in real applications.

For imbalanced data sets when the minority class is the important class, performance metrics borrowed from the information retrieval community [26] are often used. They are based on a *confusion matrix* (see Table 2) that reports the number of true positives (**TP**), true negatives (**TN**), false positives (**FP**), and false negatives (**FN**). These are then used to define metrics that evaluate the performance of a learner on the minority class, such as *recall*, *precision*, and F_β*-measure*. The formulas for these metrics are given below. The precision of a class (Eq. 6) is the number of TPs divided by the total number of examples predicted as positive. A precision score of 1.0 means that every example predicted as a positive example *is* a positive example, though there may be some positive examples that were labeled as negative. The recall of a class (Eq. 5) is the number of TPs divided by the number of examples that are actually positive. A recall score of 1.0 means that every positive example is labeled correctly, though some negative examples may have also been labeled as positive. There is always a trade-off between precision and recall, but for data sets where the cost of false negatives is high, a high recall value is preferable. The F_β-measure [26] (Eq. 8) is the weighted harmonic mean of precision and recall and merges recall and precision into a single value. The best F_β score is 1 and the worst is 0. The β parameter controls the relative weight given to recall and precision. F_β "measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision" [26]. If correct classification of the minority class is important, when false negatives have similar costs to false positives, then the F_1-measure ($\beta = 1$) is used because precision and recall are weighted equally. When the cost of false negatives is more than that of false positives, then the F_2-measure ($\beta = 2$) is better because it weights recall twice as heavily as precision.

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{7}$$

$$F_\beta = (1 + \beta^2)\frac{Recall \times Precision}{\beta^2 \times Precision + Recall} \tag{8}$$

4.2 Methodology

Of the twenty-one data sets we have available (twenty UCR data sets and the MPEG-7 data set), only three have training sets that contain a true minority class (a two-class data set with one class comprising less than 35% of the total number of examples). These data sets

Table 3 The results of adding ghost points to the OSB distance scores on the imbalanced UCR time series data sets

Data set	#GP added per minority	Overall accuracy		F_1-measure: Minority class		F_2-measure: Minority class	
	Example	SVM (%)	SVM-GP (%)	SVM	SVM-GP	SVM	SVM-GP
SyntheticControl	2	98.83	99.78	0.967	0.993	0.984	0.991
CBF	1	96.89	98.56	0.950	0.978	0.928	0.966
FaceAll	1	98.83	99.26	0.906	0.940	0.931	0.939
OSULeaf	2	86.16	87.05	0.369	0.532	0.329	0.492
SwedishLeaf	8	98.27	99.11	0.855	0.938	0.814	0.940
50Words	1	98.78	98.95	0.324	0.466	0.278	0.416
Trace	2	91.50	96.75	0.792	0.934	0.748	0.930
TwoPatterns	2	99.78	99.96	0.995	0.999	0.993	0.999
Wafer	5	96.25	99.81	0.791	0.991	0.706	0.994
FaceFour	1	91.19	96.88	0.790	0.939	0.736	0.923
Lightning2	1	73.77	83.61	0.619	0.800	0.516	0.746
Lightning7	1	89.63	93.54	0.452	0.723	0.397	0.692
ECG	1	87.00	93.00	0.787	0.896	0.710	0.857
Adiac	3	98.07	98.29	0.442	0.625	0.377	0.576
Fish	3	94.86	97.39	0.755	0.907	0.686	0.889
Beef	1	82.67	81.33	0.167	0.342	0.167	0.310
OliveOil	1	91.11	94.44	0.571	0.745	0.543	0.702

are Wafer, Lightning-2, and ECG. In order to evaluate ghost points further, we also create artificially imbalanced data sets. To create artificial minority classes for the fourteen data sets from the UCR repository that have more than two classes, we take each class that comprises less than 35% of the total number of examples as a minority class, and then collapse the remaining classes into one. If in a data set there is more than one class that meets our criteria as a minority class, we treat each class as minority class in turn and average the results. For the MPEG-7 data set, to create a training and testing set, we randomly choose ten shapes from each class (for a total of 700 shapes) for the training set, and the remaining ten shapes from each class (again 700 shapes total) become the testing set. Then we take each class as a minority class in turn, collapse the remaining 69 classes into one class, and average the results over the 70 minority classes. See Tables 3, 4, and 5 for a summary of the results.

Once we create a minority class in the training set, we add ghost points to the minority class of the training set and perform classification in the following manner:

1. The training set
   (a) Given a training set consisting of m time series examples with sequence length s, create the $m \times m$ distance matrix by calculating the OSB or DTW distance between each pair of examples.
   (b) For each minority class example x, add k-many ghost points by inserting one ghost point between x and each of its $k$nn. This gives us a total of p new points.
   (c) Calculate the distance from the p ghost points to every other point in the training set using Eqs. 1, 2, 3, or 4; we now have an $(m + p) \times (m + p)$ matrix.

Table 4 The results of adding ghost points to the DTW distance scores on the imbalanced UCR time series data sets

Data set	#GP added per minority	Overall accuracy		F_1-measure: Minority class		F_2-measure: Minority class	
	Example	SVM (%)	SVM-GP (%)	SVM	SVM-GP	SVM	SVM-GP
SynthericControl	1	97.44	99.28	0.929	0.979	0.968	0.981
CBF	1	95.83	97.72	0.934	0.964	0.917	0.945
FaceAll	8	96.08	97.56	0.731	0.844	0.792	0.837
OSULeaf	2	85.12	86.98	0.345	0.478	0.309	0.432
SwedishLeaf	9	97.94	98.71	0.829	0.907	0.791	0.911
50Words	1	98.76	98.97	0.311	0.472	0.272	0.417
Trace	2	90.25	95.50	0.769	0.909	0.717	0.899
TwoPatterns	2	98.45	99.08	0.968	0.981	0.953	0.970
Wafer	5	96.82	99.69	0.830	0.986	0.759	0.988
FaceFour	1	83.52	92.33	0.515	0.835	0.464	0.790
Lightning2	1	77.05	83.61	0.682	0.792	0.586	0.720
Lightning7	2	90.02	90.80	0.441	0.588	0.421	0.581
ECG	2	82.00	84.00	0.710	0.742	0.647	0.676
Adiac	3	97.74	98.05	0.419	0.626	0.389	0.622
Fish	5	93.55	95.76	0.708	0.845	0.650	0.824
Beef	1	82.00	81.33	0.167	0.308	0.167	0.300
OliveOil	1	85.56	91.11	0.400	0.726	0.371	0.725

(d) Convert both the original and augmented OSB or DTW score matrix to affinity matrices using the approach in [35] and Eq. 9.
(e) Use these affinity matrices as the *user-defined* or *precomputed* kernels for the SVM to get two models: one that includes ghost points and one that does not.
(f) Run SVM to train.

2. The testing set

(a) Given a testing set consisting of n time series examples with sequence length s, and a training set consisting of m time series of length s, create the $n \times m$ OSB or DTW distance score matrix.
(b) Calculate the distance from each test data point to each of the p ghost points using Eqs. 1, 2, 3, or 4; we now have an $n \times (m + p)$ distance matrix.
(c) Convert both the original and augmented OSB or DTW score matrix to an affinity matrix as in step 1d above.
(d) Use these affinity matrices as the *user-defined* or *precomputed* kernels for the SVM as in step 1e above.
(e) Run SVM to test.

There are two critical parameters to set when we convert the distance matrices to kernels that modify the σ for the Gaussian Kernel function, A and K. As stated in [37], the scaling parameter σ is some measure of when two points are considered similar. We use the method in [35] to calculate the local scaling parameter σ_{ij} for each pair of data points x_i and x_j. The

Table 5 The results of adding ghost points to the OSB and DTW distance scores on the MPEG-7 data set

Distance measure	Characteristics			Overall accuracy		F_1-measure: Minority class		F_2-measure: Minority class	
	#GP added per minority example	Number of minority example	Number of majority examples	SVM (%)	SVM-GP (%)	SVM	SVM-GP	SVM	SVM-GP
OSB	4	10	1390	99.43	99.74	0.710	0.897	0.662	0868
DTW	5	10	1390	99.11	99.20	0.603	0.767	0.581	0.794

affinity between a pair of points can be written as:

$$k(x_i, x_j) = \exp\left(\frac{-\mathrm{d}(x_i, x_j)^2}{\sigma_{ij}}\right) \tag{9}$$

where $\sigma_{ij} = A \cdot mean\{k\mathrm{nn}\,\mathrm{d}(x_i), k\mathrm{nn}\,\mathrm{d}(x_j)\}$, $mean\{k\mathrm{nn}\; d(x_i), k\mathrm{nn}\; d(x_j)\}$ is the mean distance of the K-nearest neighbors of points x_i, x_j, and A is an extra scaling parameter. For the SVM, there is a third parameter to set, which is the cost parameter C. For all UCR experiments we used $A = 0.5$, $K = 5$, and $C = 0.5$ and for all MPEG-7 experiments we used $A = 0.36$, $K = 25$, and $C = 0.5$. For each of the eighteen data sets, we run SVM on the four matrices (after converting them to kernels): OSB score matrix without ghost points; OSB score matrix with ghost points; DTW score matrix without ghost points; and DTW score matrix with ghost points.

The final parameter to set is the number of ghost points to add per minority example, as the final results can be sensitive to the number of ghost points added. Two good heuristics are 1. to balance the classes and 2. add one ghost point per minority example, but neither of these always give the best results. The strategy we use in our experiments is a modified version of balancing the classes. Let m be the number of minority examples in the training set, n be the number of majority examples, and k be the maximum number of ghost points to add per minority example such that $k \cdot m = n$. We then choose the number of ghost points per example $g \in \{1, \ldots, k\}$ that gives the best results. Though the strategy for finding g is very simple and the results are very good, g is found empirically. How to choose the *optimal* number of ghost points is an open question that we will be addressing in the future.

4.3 Results

Of the eighteen data sets we test, only three of the training sets had natural minority classes. For the other fifteen, we created artificial minority classes, and if necessary, averaged the results. See Sect. 4.2 for the methodology we used to create the imbalanced data sets. We compare the results of SVM on OSB with and without ghost points on the UCR data sets in Table 3, the results of SVM on DTW with and without ghost points on the UCR data sets in Table 4, and finally, the results of SVM on OSB and DTW on the MPEG-7 data set in Table 5. Because we are interested in the performance on minority classes, specifically minimizing the number of false negatives, we measure the overall accuracy (Eq. 7), the F_1-measure (Eq. 8 with $\beta = 1$) which weights precision and recall equally, and the F_2-measure (Eq. 8 with $\beta = 2$) which weights recall twice as heavily as precision.

As the results show in Table 3, for the OSB score matrix on the UCR data sets, adding ghost points improve SVM's overall accuracy rate on sixteen of the seventeen data sets. In fact, four of the data sets, Trace, FaceFour, Lightning-2, and ECG, have increases in overall accuracy of over 5 percentage points. On all seventeen of the data sets data sets, the F_1-measure and the F_2-measure improve with ghost points by as much as 29.5 percentage points; twelve data sets see an increase of at least 10 percentage points in the F_1-measure and thirteen data sets in the F_2-measure. For the Lightning-7 data set, adding ghost points increases the accuracy by 3.9 percentage points, the F_1-measure by 27.1 percentage points, and the F_2-measure by 29.5 percentage points. The overall accuracy of Lightning-2 has the largest increase when ghost points are added, an increase of 9.8 percentage points, while the F_1-measure and F_2-measure increase by 18.1 and 23 percentage points, respectively. The Beef data set, which is the only data set in Table 3 that decreases in overall accuracy when ghost points are added (by 1.3 percentage points), still gains in the F_1-measure, which increases by 17.5 percentage points, and the F_2-measure, which increases by 14.4 percentage points.

When using the DTW score matrix of the UCR data sets (Table 4), adding ghost points increases the overall accuracy again on sixteen of the seventeen data sets; on the Beef data set, the accuracy decreased by 0.7 percentage points. For four of the data sets (Trace, FaceFour, Lightning-2, and OliveOil), ghost points increase the accuracy by over 5 percentage points. The F_1-measure and the F_2-measure increase for all seventeen data sets when ghost points are added. Twelve data sets see an increase of at least 10 percentage points in both the F_1-measure and the F_2-measure. Two data sets (FaceFour and OliveOil) see an increase of over 30 percentage points in both their F_1-measure and the F_2-measure. The data set with the largest gain in its F_1-measure F_2-measure is OliveOil; it gains 32.6 and 35.4 percentage points, respectively. The accuracy rate for OliveOil also increases by 5.6 percentage points with ghost points. The data set FaceFour, which has the greatest accuracy gain (8.8 percentage points) also has an increase in its F_1-measure of 32 percentage points and in its F_2-measure of 32.6 percentage points. The only data set, Beef, where ghost points actually decrease the overall accuracy (by 0.7 percentage points), still has an impressive gain in the F_1-measure and the F_2-measure; 14.2 and 13.4, respectively.

For the MPEG-7 data set (Table 5), the results are similar to those discussed earlier. With both OSB and DTW, all measures increase. With OSB on the MPEG-7 data set and ghost points, the overall accuracy increases 0.3 percentage points, the F_1-measure by 18.8 percentage points, and the F_2-measure by 20.6 percentage points. The increases of the results for DTW on the MPEG-7 data set are similar; overall accuracy increases by 0.1 percentage points, the F_1-measure by 16.5 percentage points, and the F_2-measure by 21.3 percentage points.

As discussed in Sect. 2, computing the distance of a ghost point to the other points in the distance matrix can take one of three forms (see Cases 2.1, 2.2, and 2.3). Tables 6 and 7 show the number of the different types of computations made for the eighteen data sets using OSB and DTW, respectively. It is interesting to note that most of the distance spaces induced by DTW contain very few Type 2 and Type 3 computations. Fifteen of the data sets had less than 8% of non-Type 1 computations, and one of these (OliveOil) had 0%. This indicates that the distance space induced by DTW on these fifteen data sets is very close to a metric space. The remaining three data sets (Trace, Fish, and Beef) had non-Type1 computations of 33, 18%, and 44%, respectively, and thus have distance spaces relatively close to a metric space. On the other hand, the distance spaces induced by OSB are much more variable, where the number of non-Type 1 computations ranges from 6 to 85%. For example, under DTW, the OliveOil data set has 100% Type 1 computations, while only 66% under OSB; the data set Wafer has 99% Type 1 computations under DTW, but only 42% under OSB. Thus, OSB is more likely to induce non-metric distance spaces. Though again we state, and our experimental results show, that the application of the proposed method can densify distance spaces that are non-metric, and such spaces are common in many cognitively motivated tasks.

It is clear to see that ghost points increase the overall accuracy for most data sets, and also the F_1-measure and F_2-measure, at times very significantly. When a data set is imbalanced, and the cost of false negatives is high (but can't be easily quantified), then adding ghost points may significantly reduce the number of false negatives while at the same time increase overall accuracy.

5 Conclusions

We introduce an innovative method for over-sampling the minority class of imbalanced data sets. Unlike other feature based methods, our synthetic points, which we call ghost points,

Table 6 Using OSB to induce the distance matrices on the eighteen data sets tested, we count the types of distance computations made from the ghost points to every other point in the distance space (see Sect. 2). Note that due to averaging over multiple minority classes and rounding, some numbers do not add up

Data set	#GP added per minority example	Number of type 1 dist. comp.	Number of type 2 dist. comp.	Number of type 3 dist. comp.	Total number of distance computations per minority class	Percentage of type 1 (%)	Percentage of type 2 (%)	Percentage of type 3 (%)
SyutheticControl	1	29,389	10	1,826	31,225	94	0	6
CBF	1	7,796	4	607	8,407	93	0	7
FaceAll	1	67,809	48	22,923	90,780	75	0	25
OSULeaf	2	26,738	42	5,126	31,905	84	0	16
SwedishLeaf	7	182,556	261	107,177	289,993	63	0	37
50Words	1	5,014	7	3,360	8,381	60	0	40
Trace	2	1,871	34	9,352	11,256	17	0	83
TwoPatterns	2	1,500,368	270	1,124,450	2,625,087	57	0	43
Wafer	6	1,840,370	2,577	2,495,572	4,338,519	42	0	58
FaceFour	1	551	4	134	689	80	1	19
Ligbtning2	1	1,828	16	766	2,610	70	1	29
Ligbtuiug7	1	741	9	734	1,484	50	1	49
ECG	1	4,229	7	2,429	6,665	63	0	36
Adiac	3	15,615	126	9,467	25,208	62	1	38
Fish	3	13,706	65	15,282	29,053	47	0	53
Beef	1	56	17	301	375	15	5	80
OliveOil	1	235	2	118	355	66	1	33
MPEG-7	4	51,195	23	5,561	56,780	90	0	10

Table 7 Using DTW to induce the distance matrices on the eighteen data sets tested, we count the types of distance computations made from the ghost points to every other point in the distance space (see Sect. 2). Note that due to averaging over multiple minority classes and rounding, some numbers do not add up

Data set	#GP added per minority example	Number of type 1 dist. comp.	Number of type 2 dist. comp.	Number of type 3 dist. comp.	Total number of distance computations per minority class	Percentage of type 1 (%)	Percentage of type 2 (%)	Percentage of type 3 (%)
SyntheticControl	1	30,982	0	243	31,225	99	0	1
CBF	1	8,337	0	70	8,407	99	0	1
FaceAll	6	549,721	13	18,946	568,680	97	0	3
OSULeaf	2	31,609	1	295	31,905	99	0	1
SwedishLeaf	7	267,539	43	22,411	289,993	92	0	8
50Words	1	7,784	0	597	8,381	93	0	7
Trace	2	7,504	15	3,738	11,256	67	0	33
TwoPatterns	2	2,544,129	13	80,946	2,625,087	97	0	3
Wafer	5	3,566,072	274	25,564	3,591,910	99	0	1
FaceFour	1	688	0	2	689	100	0	0
Lightning2	1	2,591	0	19	2,610	99	0	1
Lightuitg7	2	3,055	0	29	3,085	99	0	1
ECG	2	14,003	0	288	14,291	98	0	2
Adiac	3	22,581	101	2,526	25,208	90	0	10
Fish	2	15,353	3	3,381	18,738	82	0	18
Beef	1	209	0	166	375	56	0	44
OliveOil	1	355	0	0	355	100	0	0
MPEG-7	4	71,101	22	103	71,225	100	0	0

are added in distance space. In addition, ghost points can be added to distance spaces that are not metric, such as those induced by elastic sequence matching algorithms like DTW and OSB. The experimental results on standard time series data sets from varied domains show that adding ghost points to the minority class can significantly improve the overall accuracy, and especially the F_1-measure and F_2-measure.

We also introduce a way to use ghost points to visualize distance data for imbalanced data sets. When plotting the distance space, adding ghost points to the minority class may change the underlying structure of the distance space such that the previously indistinct minority class now becomes observable.

We are still exploring optimal strategies for inserting ghost points. In particular, to choose the optimal number of ghost points is an open question that we will be addressing in the future.

References

1. Aach J, Church GM (2001) Aligning gene expression time series with time warping algorithms. Bioinformatics 17:495–508
2. Aizerman MA, Braverman EA, Rozonoer L (1964) Theoretical foundations of the potential function method in pattern recognition learning. In: Automation and remote control, vol 25, pp 821–837
3. Akbani R, Kwek S, Japkowicz N (2004) Applying support vector machines to imbalanced datasets. In: Proceedings of ECML'04, pp 39–50
4. Batista GEAPA, Prati RC, Monard MC (2004) A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explor. Newsl. 6(1):20–29
5. Berndt DJ, Clifford J (1994) Using dynamic time warping to find patterns in time series. In: KDD Workshop, pp 359–370
6. Bishop CM (2007) Pattern recognition and machine learning (Information Science and Statistics), 1st ed. 2006. corr. 2nd printing edn, Springer
7. Chan P, Stolfo SJ (1998) Toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection. In: In Proceedings of the fourth international conference on knowledge discovery and data mining. AAAI Press, pp 164–168
8. Chawla NV, Bowyer KW, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
9. Chawla NV, Lazarevic A, Hall LO , Bowyer KW (2003) Smoteboost: improving prediction of the minority class in boosting. In: Proceedings of the principles of knowledge discovery in databases, PKDD-2003, pp 107–119
10. Cieslak DA, Chawla NV (2008) Start globally, optimize locally, predict globally: improving performance on imbalanced data. In: 'ICDM'08: Proceedings of the 2008 eighth IEEE international conference on data mining', IEEE Computer Society, Washington, DC, USA, pp 143–152
11. Georgiou C, Hatami H (2008) CSC2414- Metric embeddings. Lecture 1: A brief introduction to metric embeddings, examples and motivation'
12. Giorgino T (2009) Computing and visualizing dynamic time warping alignments in R: the dtw package. Journal of Statistical Software 31(7):1–24
13. Han H, Wang W, Mao B (2005) Borderline-smote: a new over-sampling method in imbalanced data sets learning, vol 3644 of Lecture Notes in Computer Science, Springer, pp 878–887
14. Hovsepian K, Anselmo P, Mazumdar S (2010) Supervised inductive learning with LotkaVolterra derived models. Knowl Inf Syst
15. Keogh E, Xi X, Wei L, Ratanamahatana CA (2006) Ucr time series classification/clustering page, Website. http://www.cs.ucr.edu/~eamonn/time_series_data/
16. Kubat M, Holte RC, Matwin S (1998) Machine learning for the detection of oil spills in satellite radar images. Machine Learning 30(2–3):195–215
17. Latecki LJ, Wang Q, Köknar-Tezel S, Megalooikonomou V (2007) Optimal subsequence bijection. IEEE International conference on data Mining, pp 565–570
18. Latecki LJ, Lakaemper R, Eckhardt U (2000) Shape descriptors for non-rigid shapes with a single closed contour. In: IEEE Conference on computer vision and pattern recognition (CVPR), pp 424–429
19. Laub J, Müller K-R (2004) Feature discovery in non-metric pairwise data. J Mach Learn Res 5:801–818
20. Matousek J (2002) Lectures on Discrete Geometry. Springer-Verlag New York Inc., Secaucus, NJ, USA

21. Mena L, Gonzalez J (2006) Machine learning for imbalanced datasets: Application in medical diagnostic. In: In proceedings of the 19th international FLAIRS conference
22. Roweis S, Saul L (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290:2323–2326
23. Sakoe H, Chiba S (1978) Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans Acoust Speech Signal Process 26:43–49
24. Tenenbaum JB, de Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290:2319–2323
25. Tufte ER (2001) The visual display of quantitative information, 2nd edition. Graphics Press, Cheshire, CT
26. van Rijsbergen C (1979) In: Information retrieval. Butterworths, London
27. Vapnik VN (1995) The nature of statistical learning theory. Springer-Verlag New York Inc., New York, NY, USA
28. Wang BX, Japkowicz N (2009) Boosting support vector machines for imbalanced data sets. Knowl Inf Syst
29. Weber M, Alexa M, Müller W (2001) Visualizing time-series on spirals, In: Proceedings of the IEEE symposium on information visualization 2001 (INFOVIS'01), pp 7–14
30. Weiss GM (2004) Mining with rarity: a unifying framework. SIGKDD Explor Newsl 6(1):7–19
31. Weiss GM, Hirsh H (1998) Learning to predict rare events in event sequences. In: In Proceedings of the fourth international conference on knowledge discovery and data mining, AAAI Press, pp 359–363
32. Weiss GM, Provost F (2003) Learning when training data are costly: the effect of class distribution on tree induction. J Artif Intell Res 19:315–354
33. Woods K, Doss C, Bowyer K, Solka J, Priebe C, Kegelmeyer P (1993) Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography. Int J Pattern Recognit Artif Intell 7:1417–1436
34. Wu G, Chang EY (2003) Class-boundary alignment for imbalanced dataset learning. In: Workshop on learning from imbalanced datasets in international conference on machine learning (ICML)
35. Yang X, Bai X, Latecki LJ, Tu Z (2008) Improving shape retrieval by learning graph transduction. In: 'ECCV (4)', Vol 5305 of Lecture Notes in Computer Science, Springer, pp 788–801
36. Yi BK, Jagadish HV, Faloutsos C (1998) Efficient retrieval of similar time sequences under time warping. In: Proceedings of international conference on data engineering (ICDE98), pp 201–208
37. Zelnik-Manor L, Perona P (2004) Self-tuning spectral clustering. In: Advances in neural information processing systems 17. MIT Press, pp 1601–1608
38. Zhao H (2008) Instance weighting versus threshold adjusting for cost-sensitive classification. Knowl Inf Syst 15(3):321–334

Author Biographies

Suzan Köknar-Tezel is a PhD candidate in the Department of Computer and Information Sciences at Temple University in Philadelphia, USA. She is currently working on her PhD thesis under the supervision of Dr. Longin Jan Latecki. She received her B.S. degree and M.S. degree in computer science from Saint Joseph's University in Philadelphia in 1985 and 1993, respectively. In addition to being a PhD student, she is a visiting instructor in the Department of Computer Science at Saint Joseph's University, a position she has held since 2001. Her current research interests include classification of imbalanced data sets and time series and sequence similarity. She is a member of the ACM and a student member of IEEE.

Longin Jan Latecki received the PhD degree in computer science from Hamburg University, Germany, in 1992. He is an associate professor of computer science at Temple University, Philadelphia. His main research interests include shape representation and similarity, robot perception, and digital geometry and topology. He has published more than 175 research papers and books. He is an editorial board member of *Pattern Recognition* and *Journal of Mathematical Imaging and Vision*. He received the Pattern Recognition Society Award together with Azriel Rosenfeld for the best article published in *Pattern Recognition* in 1998. He is the recipient of the 2000 Olympus Prize, the main annual award, from the German Society for Pattern Recognition (DAGM). He is a senior member of the IEEE and the IEEE Computer Society.