ICDM 2009

Improving SVM Classification on Imbalanced Data Sets in Distance Spaces

Suzan Köknar-Tezel
Department of Computer Science
Saint Joseph's University
Philadelphia, USA
tezel@sju.edu

Longin Jan Latecki
Dept. of Computer and Information Sciences
Temple University
Philadelphia, USA
latecki@temple.edu

***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. 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 many cases also improves the overall accuracy of SVM.**

***Keywords*-imbalanced data sets; support vector machines; time series;**

I. 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 [1], credit card fraud detection [2], medical diagnostics [3], and predicting telecommunication equipment failure [4]. In these data sets, there are many examples of the "normal" (the majority/negative 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 towards 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 [5]–[7], 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]–[11]) 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 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 vector 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 using ghost points is that they 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. Fig. 1 shows the Wafer training set before and after adding ghost points. The training set has 903 examples of the majority class and 97 examples of the minority class for a total of 1000 examples. To create Fig. 1a, we first took the original 1000×1000 distance matrix and used PCA to reduce the dimensionality. For Fig. 1b we then add 9 ghost points per minority example to the distance matrix (to create a 1873×1873 matrix) and again run PCA. The majority class is plotted as red circles, the minority class as blue squares, and the ghost points as green squares. In Fig. 1a, 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. 1b, after ghost points are added to the training set, the underlying shape of the data changes to form five discernable clusters. It is clear that two of the clusters belong to the minority class (the upper-left cluster and the lower-right cluster).

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) [12], [13] and Optimal Subsequence Bijection (OSB) [14]. For a nice overview of elastic sequence matching algorithms, see [15]. 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 [16], [17]. However, DTW is particularly sensitive to outliers, since it is not able to skip any elements of the sequences. In DTW, each element of the query sequence must correspond to some element of the target sequence and vice versa. 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. 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 chose support vector machines (SVMs) to perform the

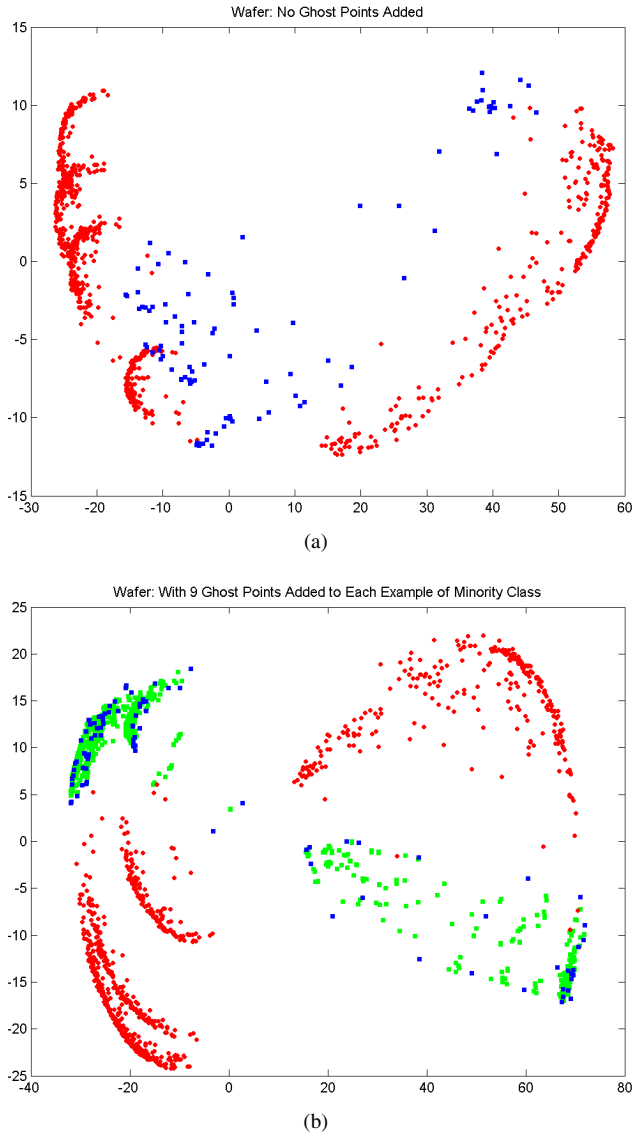

Figure 1. After using PCA on the distance matrix to reduce the dimensionality from 1000 to 2, (a) the 1,000 examples are plotted (the majority class as red circles and the minority class as blue squares). (b) is the same data set with 9 ghost points (green squares) added per minority example. In (a), 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

classification because they are a fundamental machine learning tool and they have a strong theoretical foundation [18]. 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 towards the majority/negative class [19]. 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

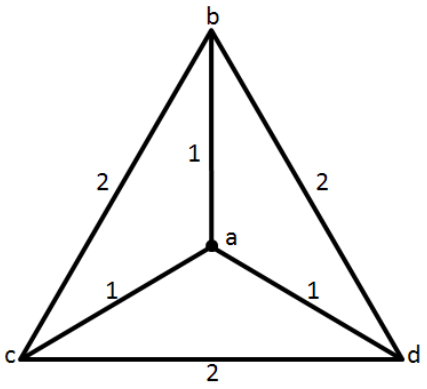

Figure 2. Example of a 4-point metric space that cannot be embedded into a Euclidean space

SVMs, but often, the misclassification costs are unknown. Our experimental results (see Sec. IV) 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 Section II, we introduce the definition of ghost points. We discuss evaluating performance on imbalanced data sets in Section III. Section IV presents our experimental results. In Section V we summarize and discuss our future work.

II. 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 if the data points do have coordinates, the Euclidean distance does not reflect their structure. Consequently, a distance measure is used that is not equivalent to the Euclidean distance, e.g., [12], [14]. For this kind 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]. A simple example where distances are not preserved when mapping to $\mathbb{R}^k$ is presented in [21]. Given the metric space X defined in Fig. 2, and the mapping $\rho : X = \{a, b, c, d\} \rightarrow \mathbb{R}^k$ for some k where ρ preserves the distances, the triangle inequality holds for the elements a, b, and d and thus the mapped points are collinear in the space $\mathbb{R}^k$. This also holds for elements a, c, and d, i.e., they are collinear in $\mathbb{R}^k$. But then $||\rho(b) - \rho(c)||_2 = 0$ contradicting the fact that the original distance between b and c is 2.

Instead of embedding, we propose to add synthetic data points directly to a given distance space. In this paper, a distance space is a pair (X, ρ), where X is a set and $\rho : X \times X \rightarrow \mathbb{R}$ is a distance function. We require only positivity, $\rho(x, y) \geq 0$ for all $(x, y) \in X \times X$, and symmetry, $\rho(x, y) = \rho(y, x)$ for all $(x, y) \in X \times X$. 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 [22].

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 $h : \Delta \rightarrow \mathbb{R}^2$ by setting $h(a) = (0, 0)$ and $h(b) = (\rho(a, b), 0)$. Then $h(x)$ is uniquely defined as a point with nonnegative coordinates such that its Euclidean distance to $h(a)$ is $\rho(x, a)$ and its Euclidean distance to $h(b)$ is $\rho(x, b)$. $h : \Delta \rightarrow \mathbb{R}^2$ 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 we do not require that (X, ρ) is a metric space, but we require that the three point space (Δ, ρ) is 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, 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 $\mu(a, b)$ in a distance space X, we need to specify $\rho(x, \mu(a, b))$ for every $x \in X$. We first isometrically embed the three point metric subspace $\Delta = \{x, a, b\} \subseteq X$ 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 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 correct distances from $\mu(a, b)$ to all points in X.

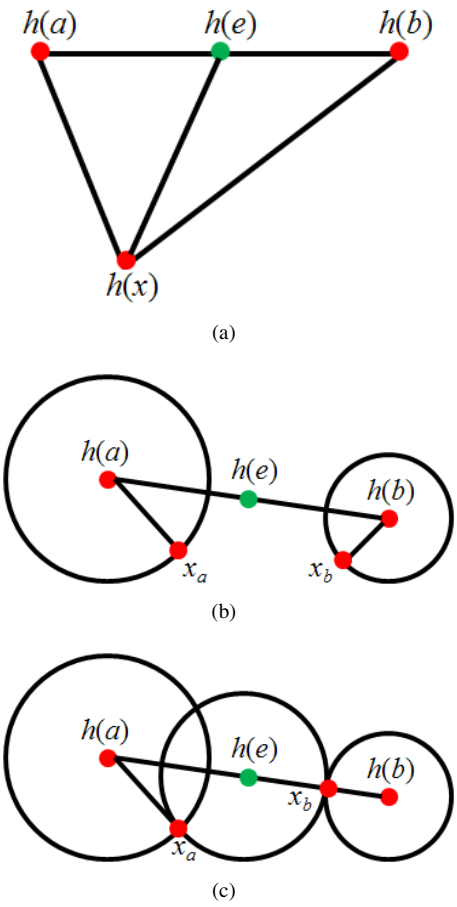

Figure 3. (a) shows the the construction of $\rho(x, e)$ for $e = \mu(a, b)$ for a triple of points that satisfy the triangle inequality. The triple of points in (b) cannot construct a triangle. The way to calculate $\rho(x, e)$ for (b) is shown in (c)

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 $y \in X$, $M_\rho(x, y) = \rho(x, y)$. 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 Eq. 1.

In Eq. 1 we assumed 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 [23], [24]. However, for a relatively small fraction of point triples, it happens for some triples $\Delta = \{x, a, b\} \subseteq X$ that (Δ, ρ) is not a metric space, which may lead to a negative distance in Eq. 1. This is the case if $\rho(x, a) + \rho(x, b) < \rho(a, b)$. Then a triangle with vertices $h(a), h(b), h(x)$ 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. Thus, the proposed approach can be applied to non-metric distance spaces, and our construction guarantees that the distances to all ghost points are nonnegative. The obtained distances also preserve symmetry.

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. 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.

	Predicted Positive	Predicted Negative
Actual Positive	**TP**	**FN**
Actual Negative	**FP**	**TN**

Table I
CONFUSION MATRIX

III. EVALUATING PERFORMANCE

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 [25], which consists of non-calcification (non-cancerous) and calcification (cancerous) examples. The data set has $11,183$ 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 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.

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 Sec. IV). 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, 1,000:1, or even 10,000:1. 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 *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 [27]. Another 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.

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 I), 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. 4) 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. 3) 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. 6) 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.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5}$$

$$F_\beta = (1 + \beta^2)\frac{Recall \times Precision}{\beta^2 \times Precision + Recall} \tag{6}$$

IV. Experimental results

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 Dynamic Time Warping (DTW) and Optimal Subsequence Bijection (OSB) as distance measures on the UCR Time Series data sets [28] for our experiments. The UCR Time Series data repository has available 20 data sets from various domains. The time series length ranges from 60 (Synthetic Control data set) to 637 (Lightning-2 data set) 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 (Face(four) data set) to 1,000 (Wafer), and the number of testing examples ranges from 28 (Coffee) to 6,174 (Wafer). In our experiments, we use fourteen of the data sets (see Fig. 4 and Fig. 6 for a list of the data sets used).

A. Methodology

We add ghost points to the minority class of the training and testing set and perform classification in the following way:

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.

	Characteristics				Overall Accuracy		F_1-Measure: Minority Class		F_2-Measure: Minority Class	
Data Set	#GP Added Per Minority Example	Number of Classes	Number of Minority Examples	Number of Non-Minority Examples	SVM	SVM-GP	SVM	SVM-GP	SVM	SVM-GP
OSULeaf	1	6	15	185	0.550	0.591	0.200	0.579	0.152	0.514
Wafer	7	2	97	903	0.964	0.999	0.802	0.994	0.719	0.996
Lightning2	1	2	20	40	0.738	0.836	0.636	0.800	0.547	0.786
ECG	2	2	31	69	0.870	0.920	0.794	0.892	0.731	0.907

Figure 4. The results of adding ghost points to the OSB distance scores on originally imbalanced time series data sets. Grayed results indicate best performers

	Characteristics				Overall Accuracy		F_1-Measure: Minority Class		F_2-Measure: Minority Class	
Data Set	#GP Added Per Minority Example	Number of Classes	Number of Minority Examples	Number of Non-Minority Examples	SVM	SVM-GP	SVM	SVM-GP	SVM	SVM-GP
OSULeaf	2	6	15	185	0.517	0.537	0.138	0.500	0.102	0.487
Wafer	5	2	97	903	0.968	0.997	0.830	0.986	0.759	0.988
Lightning2	2	2	20	40	0.770	0.820	0.682	0.792	0.586	0.766
ECG	2	2	31	69	0.800	0.790	0.688	0.696	0.640	0.678

Figure 5. The results of adding ghost points to the DTW distance scores on originally imbalanced time series data sets. Grayed results indicate best performers

c) Calculate the distance from the p ghost points to every other point in the training set using Eq. 1;we now have an $(m+p)\times(m+p)$ matrix.
d) Convert both the original and augmented OSB or DTW score matrix to affinity matrices using the approach in [29].
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, 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 Eq. 1; 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, A and K, that modify the σ for the Gaussian Kernel function. As stated in [30], the scaling parameter σ is some measure of when two points are considered similar. It is common for σ to be chosen manually, but sometimes a single value of σ does not work well for an entire data set. Therefore, we use the method in [29] to calculate the local scaling parameter σ_{ij} for each pair of data points x_i and x_j. The affinity between a pair of points can be written as:

$$k(x_i, x_j) = exp(\frac{-d(x_i, x_j)^2}{\sigma_{ij}}) \tag{7}$$

where $\sigma_{ij} = A \cdot mean\{k\text{nn } d(x_i), k\text{nn } d(x_j)\}$, $mean\{k\text{nn } d(x_i), k\text{nn } d(x_j)\}$ is the 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 experiments we used $A = 0.5$, $K = 5$, and $C = 0.5$. 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.

	Characteristics				Overall Accuracy		F_1-Measure: Minority Class		F_2-Measure: Minority Class	
Data Set	#GP Added Per Minority Example	Original Number of Classes	Number of Minority Examples	Number of Non-Minority Examples	SVM	SVM-GP	SVM	SVM-GP	SVM	SVM-GP
SyntheticControl	4	6	50	250	0.967	0.997	0.909	0.990	0.962	0.996
FaceAll	2	14	40	520	0.997	0.999	0.925	0.984	0.951	0.975
SwedishLeaf	6	15	26	474	1.000	1.000	1.000	1.000	1.000	1.000
Trace	3	4	21	79	0.930	0.970	0.863	0.945	0.797	0.915
FaceFour	1	4	3	21	0.807	0.977	0.514	0.960	0.398	0.938
Lightning7	3	7	5	65	0.877	0.959	0.182	0.824	0.122	0.745
Adiac	1	37	4	386	0.959	0.962	0.000	0.286	0.000	0.217
Fish	2	7	21	154	0.954	0.983	0.840	0.945	0.766	0.916
Beef	3	5	6	24	0.800	0.800	0.000	0.400	0.000	0.357
OliveOil	3	4	4	26	0.900	0.933	0.400	0.667	0.294	0.556

Figure 6. The results of adding ghost points to the OSB distance scores on artificially imbalanced time series data sets. Grayed results indicate best performers

	Characteristics				Overall Accuracy		F_1-Measure: Minority Class		F_2-Measure: Minority Class	
Data Set	#GP Added Per Minority Example	Original Number of Classes	Number of Minority Examples	Number of Non-Minority Examples	SVM	SVM-GP	SVM	SVM-GP	SVM	SVM-GP
SyntheticControl	1	6	50	250	0.993	1.000	0.980	1.000	0.992	1.000
FaceAll	6	14	40	520	0.988	0.996	0.837	0.945	0.826	0.923
SwedishLeaf	11	15	26	474	0.982	1.000	0.882	1.000	0.854	1.000
Trace	3	4	21	79	0.920	0.970	0.840	0.945	0.766	0.915
FaceFour	2	4	3	21	0.727	0.955	0.143	0.917	0.094	0.873
Lightning7	1	7	5	65	0.863	0.877	0.000	0.182	0.000	0.122
Adiac	2	37	4	386	0.959	0.962	0.000	0.348	0.000	0.282
Fish	2	7	21	154	0.943	0.949	0.792	0.816	0.704	0.735
Beef	2	5	6	24	0.800	0.800	0.000	0.400	0.000	0.357
OliveOil	3	4	4	26	0.867	0.933	0.000	0.667	0.000	0.556

Figure 7. The results of adding ghost points to the DTW distance scores on artificially imbalanced time series data sets. Grayed results indicate best performers

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. If the data set is highly imbalanced, a good heuristic is to balance the the classes, but this does not always give the best results. How to choose the optimal number of ghost points is an open question that we will be addressing in the future.

B. Results

Of the 20 UCR time series data sets, there are only four that have a true minority class (the smallest class is at most 50% of the size of the next smallest class). These data sets are OSU Leaf, Wafer, Lightning-2, and ECG. OSU Leaf has 6 classes, and the other three have 2 classes each. We compare the results of SVM on OSB with and without ghost points on the four data sets in Fig. 4 and the results of SVM on DTW with and without ghost points on the four data sets in Fig. 5. Because we are interested in the performance on minority classes, specifically minimizing the number of false negatives, we measure the overall accuracy (Eq. 5), the F_1-measure (Eq. 6 with $\beta = 1$) which weights precision and recall equally, and the F_2-measure (Eq. 6 with $\beta = 2$) which weights recall twice as heavily as precision.

As the results show in Fig. 4, for the OSB score matrix, adding ghost points increases, for all four data sets, the overall accuracy, the F_1-measure and the F_2-measure. Adding ghost points to the OSU Leaf data set increases the overall accuracy by only 4.1 percentage points, but the F_1-measure and the F_2-measure increase by 37.9 and 36.2 percentage points respectively. For the ECG data set, adding ghost points increases the overall accuracy by 9.8 percentage points, and the F_1-measure and the F_2-measure by 16.4 and 23.9 percentage points respectively.

When using the DTW score matrix (Fig. 5), adding ghost points increases the overall accuracy, the F_1-measure, and the F_2-measure for three of the four data sets. For OSU Leaf, adding ghost points increases the overall accuracy by only 2 percentage points, but the F_1-measure and the F_2-measure increase by 36.2 and 38.5 percentage points respectively. The only decrease in performance is for the ECG data set, where the accuracy decreases 1 percentage point when ghost points are added, but even here the F_1-measure and the F_2-measure increase.

In order to evaluate ghost points further, we also create artificially imbalanced data sets. Using ten other data sets from the UCR repository that have more than two classes, we keep the class with the least number of examples as the minority class, and then collapse the remaining classes into one, giving us imbalanced data sets with two classes each. If there is more than one "least" class, we choose randomly among them the class that will be the minority class. We then perform the same steps described above. See Fig. 6 and Fig. 7 for the results.

With the OSB score matrix (Fig. 6), ghost points improve SVM's overall accuracy rate on eight of the ten data sets (on the other two data sets, the accuracy remains the same). On nine of the ten data sets, the F_1-measure and the F_2-measure improve with ghost points. For the Face Four data set, the overall accuracy increases 17 percentage points by adding ghost points, and the F_1-measure and the F_2-measure increase 44.6 and 54 percentage points respectively. On the Lightning-7 data set, ghost points increase the overall accuracy by 8.2, the F_1-measure by 64.2, and the F_2-measure by 62.3 percentage points. For the Beef data set, though the accuracy does not change by adding ghost points, without them the F_1-measure and the F_2-measure are zero since SVM does not classify any of the six positive examples as positive. Adding ghost points increases the F_1-measure and the F_2-measure by 40 and 35.7 percentage points respectively. The Swedish Leaf data set is the only data set that does not improve in any of the performance measures because the overall accuracy, recall, and precision are 100% both with and without ghost points.

The results are similar for DTW, as seen in Fig. 7. On nine of the ten data sets, the overall accuracy improves with ghost points (on the tenth data set the accuracy is unchanged). On all ten data sets the F_1-measure and the F_2-measure increase. The most dramatic increase is on the Face Four data set. Overall accuracy increases 22.8 percentage points with ghost points, the F_1-measure increases 77.4 percentage points and the F_2-measure increases 77.9 percentage points. For four of the data sets, without ghost points the F_1-measure and the F_2-measure are zero, i.e., SVM does not classify *any* of the positive examples correctly. Adding ghost points allows the SVM to correctly identify at least some of the positive examples.

V. 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, are added in distance space. The experimental results on standard time series data sets 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. In our future work, we will explore optimal strategies for adding ghost points.

References

[1] M. Kubat, R. C. Holte, S. Matwin, R. Kohavi, and F. Provost, "Machine learning for the detection of oil spills in satellite radar images," in *Machine Learning*, 1998, pp. 195–215.

[2] P. Chan and S. J. Stolfo, "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, 1998, pp. 164–168.

[3] L. Mena and J. Gonzalez, "Machine learning for imbalanced datasets: Application in medical diagnostic," in *In Proceedings of the 19th International FLAIRS Conference*, 2006.

[4] G. M. Weiss and H. Hirsh, "Learning to predict rare events in event sequences," in *In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1998, pp. 359–363.

[5] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 20–29, 2004.

[6] D. A. Cieslak and N. V. Chawla, "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*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 143–152.

[7] G. M. Weiss, "Mining with rarity: a unifying framework," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 7–19, 2004.

[8] N. V. Chawla, K. W. Bowyer, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[9] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "Smoteboost: improving prediction of the minority class in boosting," in *Proceedings of the Principles of Knowledge Discovery in Databases, PKDD-2003*, 2003, pp. 107–119.

[10] R. Akbani, S. Kwek, and N. Japkowicz, "Applying support vector machines to imbalanced datasets," in *ECML*, 2004, pp. 39–50.

[11] H. Han, W. Wang, and B. Mao, "Borderline-smote: A new over-sampling method in imbalanced data sets learning." ser. Lecture Notes in Computer Science, vol. 3644. Springer, 2005, pp. 878–887.

[12] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *KDD Workshop*, 1994, pp. 359–370.

[13] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, pp. 43–49, 1978.

[14] L. J. Latecki, Q. Wang, S. Köknar-Tezel, and V. Megalooikonomou, "Optimal subsequence bijection," *Data Mining, IEEE International Conference on*, vol. 0, pp. 565–570, 2007.

[15] T. Giorgino, "Computing and visualizing dynamic time warping alignments in R: The dtw package," *Journal of Statistical Software*, vol. 31, no. 7, pp. 1–24, 2009.

[16] Aach and Church, "Aligning gene expression time series with time warping algorithms," *Bioinformatics*, vol. 17, pp. 495–508, 2001.

[17] Yi, Jagadish, and Faloutsos, "Efficient retrieval of similar time sequences under time warping," in *Proceedings Int. Conf. on Data Engineering (ICDE98)*, 1998, pp. 201–208.

[18] V. N. Vapnik, *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[19] G. Wu and E. Y. Chang, "Class-boundary alignment for imbalanced dataset learning," in *Workshop on Learning from Imbalanced Datasets in International Conference on Machine Learning (ICML)*, 2003.

[20] J. Matousek, *Lectures on Discrete Geometry*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2002.

[21] C. Georgiou and R. H. Hatami, "CSC2414- Metric embeddings. Lecture 1: A brief introduction to metric embeddings, examples and motivation," 2008.

[22] J. Laub and K.-R. Müller, "Feature discovery in non-metric pairwise data," *Journal of Machine Learning Research*, vol. 5, pp. 801–818, 2004.

[23] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[24] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[25] K. Woods, C. Doss, K. Bowyer, J. Solka, C. Priebe, and P. Kegelmeyer, "Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, pp. 1417–1436, 1993.

[26] C. van Rijsbergen, in *Information Retrieval*. Butterworths, London, 1979.

[27] G. M. Weiss and F. Provost, "Learning when training data are costly: The effect of class distribution on tree induction," *Journal of Artificial Intelligence Research*, vol. 19, pp. 315–354, 2003.

[28] Keogh, Xi, Wei, and Ratanamahatana, "Ucr time series classification/clustering page," Website, http://www.cs.ucr.edu/~eamonn/time_series_data/.

[29] X. Yang, X. Bai, L. J. Latecki, and Z. Tu, "Improving shape retrieval by learning graph transduction." in *ECCV (4)*, ser. Lecture Notes in Computer Science, vol. 5305. Springer, 2008, pp. 788–801.

[30] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in Neural Information Processing Systems 17*. MIT Press, 2004, pp. 1601–1608.