Learning Imbalanced Data with Manifold-based Sampling

Canh Hao Nguyen and Tu Bao Ho
School of Knowledge Science
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa, 923-1211, Japan
{canhhao,bao}@jaist.ac.jp

Abstract

Imbalanced data learning received considerable attention from the research and industrial communities. It is problematic as traditional machine learning methods fail to achieve satisfactory results due to the skewed class distribution. Solutions to the problem generally use traditional machine learners to make a bias decision in favor of the smaller class. To make such a bias decision, one usually needs a good assumption of data distributions. The commonly made assumptions are Gaussian distributions, subspaces, clusters or the implicit assumptions behind classifiers. For many imbalanced data sets, due to the shortage of small class data, distributions are hard to be recognized, making the common assumptions too strong to hold true in practice. We propose to use a more flexible assumption of the small class lying on a manifold. Based on recent advances in manifold learning algorithms, basic sampling strategies to account for skewed class distribution are designed. However, it is also shown that some interpolation-based sampling strategies suffer from several statistical drawbacks. Another sampling strategy is derived to overcome these drawbacks. The algorithms using these sampling strategies show the merit of the sampling strategies and the utility of the manifold assumption.

1 Introduction

Imbalanced dataset is one with very a skewed class distribution. For instance, in a binary classification problem, when one class accounts for only 2%, the other class has 98% of the examples. In such a situation, traditional classifiers would classify every example to the large class with an overall accuracy of 98%. However, the small class is not leant at all. The small class is outnumbered, therefore prediction for the small class would appear to be less statistical significant. In practice, the problem is encountered in various domains, for example, diagnoses of rare diseases [17], fraudulent transaction detection [5], oil spills in satellite images [9], rooftop detection [11], biological data [12], network intrusion detection [10], etc. In these examples, the small class is usually of primary interest; hence an overall accuracy of 98% does not make any sense.

The reason that traditional classifiers fail to learn the small class is that they tend to make the fundamental assumption of an equal class distribution. When dealing with imbalanced data, most approaches bend this assumption by introducing biases into traditional classification methods. The ground, on which a bias is introduced, is usually some form of data distribution. A natural assumption is that small class data is simply sparser. In this case, rebalancing class distribution, either by adding examples to small class (upsampling) or removing examples from the large class (downsampling), would be sufficient. It was observed that these sampling methods did not give a good performance [7]. Simple cost-sensitive method, which gives distinct costs to classes, does not make much difference in various classification methods [4]. Similarly, SMOTE [2] generates synthetic data to add to the small class using nearest neighbor links. It is basically relied on this assumption.

Various other assumptions have been made on the effect of imbalanced data. Elkan [4] claimed that rebalancing has little effect on decision functions, but it is more effective to use cost-sensitive learning. It was claimed the imbalanced data effect happens within clusters in the data [13]; rebalancing class distributions on clusters would solve the problem. The concept of Tomek link is used to downsample the large class [9]. Imbalanced data effect is attributed to small disjuncts, the ones with only a few small class examples [8]. Higher spatial resolution in feature space is given to small class [18] as small class is thought to occupy a small region. Various assumptions were proposed, but none have been made a de facto standard. It is a common belief that each assumption is only good for some data types. Therefore, it is necessary to introduce different assumptions and corresponding solutions for different data types. The less strict an assumption is, the more applicable it can be.

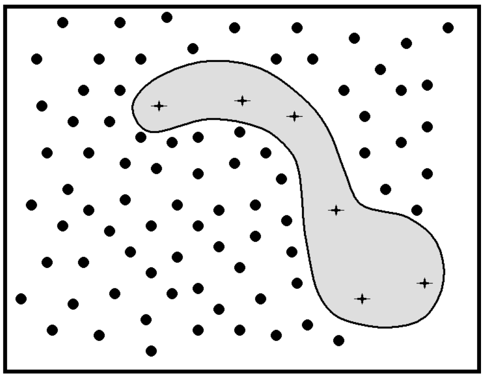

Figure 1. A manifold of small class.

In search of a good assumption for imbalanced data distribution, we used the notion of manifold. Manifold is a way to model complicated data distribution and to reveal hidden structures. The key distinction from previous approaches is that manifold is a *flexible* framework, which does not impose strict assumptions on small class data distribution. It does not require data to lie in clusters, linear subspaces or small disjuncts. Figure 1 shows an example of the small class distribution that may be difficult to characterize using traditional distribution models, and manifold is expected to be an effective alternative. Having assumed the manifold structure of small class data, we deal with the imbalanced data problem by generating synthetic examples with two sampling strategies.

Different from previous works, we show that the sampling strategies based on interpolation like SMOTE and our in-class sampling suffer from some statistical drawbacks. It is shown that those sampling strategies do not increase the convex hull of the small class data. They also decrease the variance of the small class. These drawbacks mean that those sampling strategies may not be sufficient to make a bias decision in favor of the small class. The out-class sampling strategy is designed to overcome the drawbacks of using only interpolation-based samplings. We then propose three algorithms based on the two sampling strategies. Experiments show the necessity of these strategies and the effectiveness of algorithms based on them.

In this work, we first review the fundamental idea behind manifold learning in Section 2. We then design two sampling strategies for imbalanced data in Section 3. In Section 4, we describe a family of three algorithms using the strategies. We then evaluate those algorithms in comparison with other classifiers in Section 5. We then conclude the paper with some outlooks.

2 Manifold Learning

The driving force of manifold learning is for the problem of intrinsically low dimensional data that lies in a high dimensional space. Such problems are encountered across domains various domains. The target of manifolds learning algorithms is to discover the low and meaningful dimensional representation of data [16]. The fundamental assumption in manifold learning algorithms is that data should lie on a manifold, which is viewed as a Riemannian submanifold of the ambient Euclidean space and is globally isomorphic to a convex subset of a low dimensional space [3]. In this section, we review the key idea behind the manifold assumption and borrow it for the imbalanced data problem.

Recently, a new family of manifold learning algorithms has been proposed to characterize the intrinsic geometric structure of a manifold and embed it into a low (hopefully meaningful) dimensional space. Representative algorithms are ISOMAP [16] and LLE [14]. The frameworks of these algorithms are quite similar and can be unified as: constructing a neighborhood graph and distill information, then embedding the data into a low dimensional space preserving the information. The first step in the framework extracts information characterizing the manifold. After the abstraction process, the information is used to construct a low dimensional space representation of the original manifold. Various algorithms extract different information for computational purposes, but basically, the information is based on some neighborhood graphs. The ISOMAP algorithm can be described as follows:

Given a data set $X = \{x_i\}_{i=1}^n, x_i \in R^d$, we wish to find a mapping $\phi : R^d \to R^{d'}$ such that the mapping preserves some desired information.

1. Determine which points are neighbors in the manifold based on distance between pairs of points $d(x_i, x_j)$. Two simple methods are connecting points within some fixed radius ε or connecting all k-nearest neighbors. The connections form a weighted graph G.

2. Estimate the geodesic distances $d_M(x_i, x_j)$ between all pairs of points on the manifold by the shortest path distances $d_G(x_i, x_j)$ from the graph G.

3. Use the Multidimensional Scaling method to construct an embedding of X in the lower dimensional space $y_i = \phi(x_i) \in R^{d'}$ by minimizing an objective function that tries to preserve geodesic distances.

LLE is slightly different that instead of preserving pairwise distances, it preserves linear coefficients that reconstruct each data point from its neighbors.

These algorithms directly model the manifold of data points by constructing a neighborhood graph. Information on the graph, such as geodesic distances or linear coefficients to reconstruct data points, characterizes the manifold. These algorithms are capable of modeling nonlinear manifolds. Manifold modeling is flexible in the sense that it does not make any strict assumption of data distribution like clusters, linear subspaces, Gaussian mixtures and so on. This motivates us to use manifold to model the small class in imbalanced data. The reason is that in imbalanced data, the small class is difficult to learn due to its shortage of data and may not exhibit any regularity. Therefore, the manifold assumption would be weak enough to for imbalanced data, when other common but strong assumptions fail.

3 Sampling Strategies

Having assumed that small class examples lie in a manifold, the first step in a manifold learning framework could be used to extract relevant information of the manifold to deal with imbalanced data. As the small class is short of training examples, it is expected that the manifold would be represented by an inefficient number of examples. Therefore, we use the manifold assumption to generate more synthetic training examples to add to the small class in order to account for the imbalanced data problem. We fisrt describe the in-class sampling to *enhance* the manifold structure based on interpolation operators. Then statistical drawbacks of interpolation-based samplings are presented. To overcome the drawbacks, we propose the out-class sampling to *expand* the manifold structure, enlarging the manifold region to have a bias in favor of the small class.

3.1 In-class Sampling

Our method for modeling the manifold of the small class follows the common framework of manifold learning as ISOMAP and LLE. To enhance the manifold structure, the strategy generates synthetic examples for the small class with the requirement that synthetic examples should lie in the manifold. Therefore, it is natural to choose synthetic examples as points in the line segment connecting nearest neighbors. The in-class sampling strategy is described in Figure 2.

This strategy is similar to SMOTE [2] in the sense that they both base on interpolation operators on the small class only, i.e. using nearest neighbor line segments. The strategy is different from SMOTE in the sense that it is fully deterministic, while SMOTE chooses among k-nearest neighbors randomly and generate synthetic examples randomly in the line segments. The idea of generating synthetic examples of to make data more dense was also used in [6]. The assumption behind is that line segments between nearest neighbors are likely to lie inside or near the region of the small class. This is also the assumption of most of manifold learning algorithms, to model manifold with neighborhood graphs.

```
Input: $D^+$ is set of small class examples, $x_i \in D^+$
Parameter: $k$ is number of nearest neighbors,
    $n$ is sampling degree
Output: Synthetic examples $S^+$

1. Look for $x_i$'s k-nearest neighbors in $D^+$.
    $NN^+(x_i) \subset D^+$, $|NN^+(x_i)| = k$
2. Choose from its k-nearest neighbors n examples
    with the largest distances to $x_i$.
    $nNN^+(x_i) \subset NN^+(x_i)$, $|nNN^+(x_i)| = n$
3. For each chosen neighbor, generate a synthetic example
    as the middle point of the line segment between it and $x_i$.
    $\forall x_j \in nNN^+(x_i), x_{ij} = \frac{x_i+x_j}{2} \rightarrow x_{ij} \in S^+$
```

Figure 2. In-class Sampling Strategy.

3.2 Drawbacks of Interpolation-based Sampling

In-class sampling and SMOTE add synthetic examples into the small class, making numbers of examples from classes less skewed. However, increasing the number of examples of the small class does not always have a desirable effect. In this section, we show that even having more examples, one may not have a bias decision. For some classifiers, generating more synthetic examples these ways has an opposite effect.

Theorem 1: *The synthetic examples generated by in-class sampling always lie inside the convex hull of the original small class examples.*

The proof of this property is straight from the convexity of convex hull: all line segments connecting points inside the convex hull lie entirely within the convex hull. In case the shortage of the small class data causes the shrinkage of the ideal convex hull, only in-class sampling strategy would be insufficient. For hard margin-based linear classifiers, generating synthetic examples this way does not make any bias decision.

Theorem 2: *The inclusion of synthetic examples into the original data set reduces the expected (bias-corrected) variance of small class data.*

Proof: Denote the set of small class examples as $D^+ = \{x_i\}_{i=1}^n$. Then the mean of the set is $\overline{x} = \frac{1}{n}\sum_{i=1}^n x_i$ and its (bias-corrected) variance is: $var_1 = var(D^+) = \frac{1}{n-1}\sum_{i=1}^n (x_i - \overline{x})^2$.

Denote the set of p generated synthetic examples as $S^+ = \{x_i\}_{n+1}^{n+p}$. The new mean of all small class examples now is $\overline{x'} = \frac{1}{n+p}\sum_{i=1}^{n+p} x_i$. The variance of the new small class data is $var_2 = var(D^+ \cup S^+) = \frac{1}{n+p-1}\sum_{i=1}^{n+p}(x_i - \overline{x'})^2$.

Denote $d = min\|x_i - x_j\|, 1 \leqslant i < j \leqslant n$ and $l = \|\overline{x} - \overline{x'}\|$.

The way in-class sampling generate synthetic examples is: $x_{n+m} = \frac{x_i + x_j}{2}$, then for any x, $(x_i - x)^2 + (x_j - x)^2 = 2(x_{n+m} - x)^2 + \frac{(x_i - x_j)^2}{2} \geqslant 2(x_{n+m} - x)^2 + \frac{d^2}{2}$.

If we assume that i, j are random indices in $\{1..n\}$, then the expected value of $\sum_{m=1}^{p}(x_{n+m} - \overline{x})^2 \leqslant \frac{p}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2 - \frac{p}{2}d^2$.

Then we have:

$$
\begin{aligned}
(n+p-1) * var_2 &= \sum_{i=1}^{n+p}(x_i - \overline{x'})^2 \\
&= \sum_{i=1}^{n+p}(x_i - \overline{x'})^2 + 2(\overline{x'} - \overline{x})\sum_{i=1}^{n+p}(x_i - \overline{x'}) \\
&+ (n+p)\{(\overline{x} - \overline{x'})^2 - l^2\} \\
&= \sum_{i=1}^{n+p}(x_i - \overline{x})^2 - (n+p)l^2 \qquad (1) \\
&\leqslant \frac{n+p}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2 - \frac{p}{2}d^2 - (n+p)l^2 \\
var_2 &\leqslant \frac{(n+p)(n-1)}{n(n+p-1)} * var_1 - \frac{(n+p)l^2 + \frac{p}{2}d^2}{n+p-1} \\
var_2 &< (1 - \frac{p}{n(n+p-1)}) * var_1 \\
var_2 &< var_1. \ \square
\end{aligned}
$$

As in-class sampling is a deterministic version of SMOTE, we can easily see that SMOTE, which is also based on interpolation operator, suffers from these drawbacks.

Theorems 1 and 2 show drawbacks for learning imbalanced data as for the shortage of training examples; it is reasonable to expect that the convex hull of the small class may be shrunken down. It is also expected that the variance of small class is not greater than the true one. Hence, using interpolation-based sampling would result in smaller variances than the true variance. Hence, interpolation-based samplings cannot generate the correct the true data distribution for the small class.

These statistical drawbacks may explain why it was observed by Elkan [4] that rebalancing class distributions make little different in decision functions, but it is more promising to use cost sensitive learning. The reason is that rebalancing class distribution traditionally uses interpolation-based sampling or resampling the original data sets that would suffer from the drawbacks (or at least does not give any counter effect). Furthermore, based on these drawbacks, we will show that it is possible to use appropriate sampling strategies to overcome these drawbacks. This would refute the claim from [4].

```
Input: $x_i \in D^+$ is a small class examples,
    $D^-$ is set of large class examples.
Parameter: $k$ is number of nearest neighbors,
    $n$ is sampling degree,
    $\epsilon$ is expansion degree.
Output: Synthetic examples $S^+$.

1. Look for $x_i$'s k-nearest neighbors in $D^+$.
    $NN^-(x_i) \subset D^-$,$|NN^-(x_i)| = k$
2. Choose from its k-nearest neighbors n examples
    with the smallest distances to $x_i$
    $nBN^-(x_i) \subset NN^-(x_i)$, $|nNN^-(x_i)| = n$
3. For each chosen neighbor, generate a synthetic example
    as a point in the line
    segment between it and $x_i$.
    $\forall x_j \in nNN^-(x_i), x_{ij} = (1-\epsilon)x_i + \epsilon x_j \to x_{ij} \in S^+$
```

Figure 3. Out-class Sampling Strategy.

3.3 Out-class Sampling

The previous section proves that in-class sampling does not increase the convex hull, or the (bias-corrected) variance of small class data. However, it is reasonable to think that the shortage of data for the small class may shrink down the learned manifold. It is necessary to introduce new synthetic examples to compensate for this effect and hope it better reflects an ideal small class data distribution. The effect of shrinking a manifold would move class boundary toward the small class, therefore we wish to expand the manifold toward the boundary of classes. However, detecting the boundary of classes would be hard and algorithm specific. A way around this is to look for nearest neighbors from the other classes (the large class in binary classification problems). Therefore, we expand the manifold of small class by generating synthetic examples linking each small class example to its nearest neighbors in the large class. We call this out-class sampling as in Figure 3.

As analyzed above, interpolation-based samplings suffer from those statistical drawbacks. One can easily see that generating examples using neighborhood links to the other class, we are likely to overcome those drawbacks. We can expect the enlargement of convex hulls and the increase of the variances. This trategy is expected to complement the effects caused by using interpolation-based samplings. This strategy is designed to used together with interpolation-based samplings.

By default, we set $\epsilon = \frac{1}{3}$. This means that the generated examples are at one third of the way from the small class

```
Input: $D^+$ is set of small class examples,
    $D^-$ is set of large class examples.
Parameter: $k$ is number of nearest neighbors,
    $inn$ is degree of in-class sampling,
    $outn$ is degree of out-class sampling.
Output: Synthetic examples $S^+$.

For each $x_i \in D^+$:
1. In-class sampling with sampling degree $inn$.
2. Out-class sampling with sampling degree $outn$.
```

Figure 4. Monolithic algorithm.

examples to their neighbors in the other class. The strategy generates examples in the line segment between a small class example and one of its neighbors from the large class. This will move the class boundary toward the large class and expand the small class region, overcoming the two drawbacks of in-class sampling. The way that out-class sampling moves the class boundary is different from translating decision boundary toward the large class as in [1]. Out-class sampling moves the boundary in original data space depending on nearest neighbors from the large class. The farther those nearest neighbors are, the more boundary is moved. This makes the bias of out-class sampling adaptive to each example, making it close to cost-sensitive learning strategies.

4 Manifold Sampling Algorithms

In this section, we describe three algorithms that use sampling strategies for the imbalanced data problem. The algorithms differ in the way they deploy those sampling strategies. These are: the Monolithic, Adaptive and Selective algorithms. The Monolithic algorithm simply combines in-class sampling and out-class sampling. The Adaptive algorithm uses the two sampling strategies adaptively depending on the example being considered. The Selective algorithm guesses when an example being considered needs to be sampled.

4.1 Monolithic Algorithm

A natural way to combine the two sampling strategies is to use both of them. The Monolithic algorithm uses both in-class sampling and out-class sampling for each small class example. It is summarized in Figure 4.

For each small class example, there will be $inn + outn$ synthetic examples generated around it.

```
Input: $D^+$ is set of small class examples,
    $D^-$ is set of large class examples.
Parameter: $k$ is number of nearest neighbors,
    $n$ is total degree of sampling.
Output: Synthetic examples $S^+$.

For each $x_i \in D^+$:
1.Calculate average distances to nearest neighbors.
    $pnd(x_i) = \overline{d(x_i, x_j)}, x_j \in NN^+(x_i)$
    $nnd(x_i) = \overline{d(x_i, x_j)}, x_j \in NN^-(x_i)$
    $inn = round(n * \frac{nnd}{pnd+nnd})$
    $outn = n - inn$
2. In-class sampling with sampling degree $inn$.
3. Out-class sample with sampling degree $outn$.
```

Figure 5. Adaptive algorithm.

4.2 Adaptive Algorithm

Modelling the small class with a manifold, it is reasonable to assume that some examples to lie well inside the manifold, some are on its boundary. If one is inside, it is better to increase the data density around it. On the other hand, if one is in the boudary of the manifold, it is necessary to concentrate on expanding the boundary. We present an adaptive way of generating synthetic examples, called the Adaptive algorithm, based on the following criteria:

- If a small class example is near the boundary of classes, use more out-class sampling to expand the boundary of small class.
- If a small class example is well inside the class region, use more in-class sampling to increase the density of the class region.

The algorithm is described in Figure 5. In the algorithm, there is only one parameter (except for the number of nearest neighbors), which is the total degree of sampling for each small class example. We use the relative ratio of average distance to the small and the other class to determine how close an examples to the class boundary. The algorithm calculates the degree of in-class and out-class sampling in an adaptive manner in step *1*.

4.3 Selective Algorithm

By modelling the small class with a manifold, one may expect that there are noises on the manifold. To be robust to the noise picked by manifold learning algorithms, we propose to use a simple filtering method that detect the reliable examples, the one that does not lie well inside the other

```
Input: $D^+$ is set of small class examples,
    $D^-$ is set of large class examples.
Parameter: $k$ is number of nearest neighbors,
    $inn$ is degree of in-class sampling,
    $outn$ is degree of out-class sampling.
Output: Synthetic examples $S^+$.

For each $x_i \in D^+$:
1. Calculating average distances to nearest neighbors.
    $pnd(x_i) = \overline{d(x_i, x_j)}, x_j \in NN^+(x_i)$
    $nnd(x_i) = \overline{d(x_i, x_j)}, x_j \in NN^-(x_i)$
2. If $nnd > pnd$, continue next x
3. In-class sampling with sampling degree $inn$.
4. Out-class sampling with sampling degree $outn$.
```

Figure 6. Selective algorithm.

class. The algorithm for this intuition, called Selective algorithm, is based on the following criteria:

- If an example is near the boundary of classes or in the region of its class, use both sampling strategies to enhance and expand the manifold structures.
- If an example is well inside the other class region, do not sample.

The algorithm is described in Figure 6. The algorithm has two sampling parameters, in-class and out-class sampling degrees as in the Monolithic algorithm. By default, we use a heuristics to detect the examples lying closer to the large class than the small class in step 1 and 2. The heuristics means that when an example is too close to the other class, it is not used for sampling.

4.4 Discussion

These three algorithms base on different heuristics to combine in-class and out-class samplings. Monolithic and Selective algorithms need two parameters, namely sampling degrees while Adaptive algorithm needs only one (like SMOTE). Monolithic algorithm blindly uses both in-class and out-class sampling. Adaptive algorithm uses the relative position of each example on the learnt manifold to have an appropriate sampling rates. Selective algorithm, on the other hand, chooses only the examples from the small class with high confidence to lie on the manifold of the small class to sample.

5 Experimental Evaluation

In this section, we first carried out experiments to show that interpolation-based samplings, which decrease the variances of the small class, would not have the ability to have classifiers made a bias decision in favor of small class using simple synthetic data sets. We then run on real text classification data sets to show that the algorithms we proposed actually can help to increase the ability to learn the small class in comparison with the similar technique of SMOTE. We also showed that it is also the case in practice that using out-class sampling, which increases the small class data variance, can be very significant in real situations.

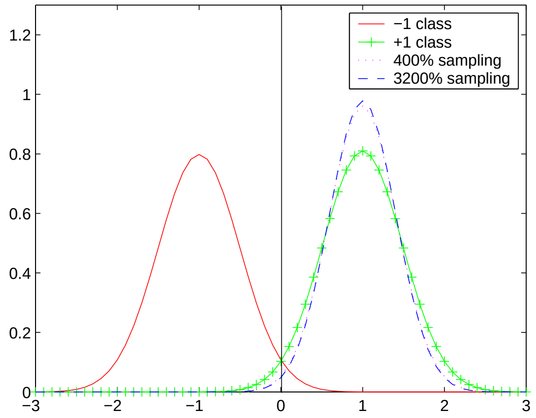

Figure 7. Gaussian distributions with different sampling rates.

5.1 Drawbacks of Interpolation-based Sampling

We simulated simple data distributions to show the drawbacks of interpolation-based sampling. We generated data for two class, $+1$ at $(1, 0) \in R^2$ and -1 at $(-1, 0) \in R^2$ according to Gaussian distribution with variance $\frac{1}{2}$. Class $+1$ has 25 examples while class -1 has 800 examples. We then applied the in-class sampling to class $+1$ repeatedly 6 times, each time with 100% sampling degrees. After that, the final data set for class $+1$ contains exactly 800 training examples. We monitored the variances of classes and graphed their (normalized Gaussian) distributions to show in one-dimensional space in Figure 7.

In the original data sets for both classes, their (bias corrected) variances are very close to 0.5 as generated. During the course of in-class sampling, the variances of class $+1$ are 0.4331, 0.4150, 0.4100, 0.4083 and 0.4077. One can see from the figure that at first, the two distributions are quite similar and close to the original data generation models. However, the variance of class $+1$ quickly decreases (the tops of the data distributions increase). Please note that at the 3200% sampling degree for the $+1$ class, numbers of training examples of the two classes are equal. However, optimal classification would make a bias decision for

Table 1. Statistics and F-measure results on Reuters-21578 data.

class	#train	%train	#test	SVMs	SMOTE	Monolithic	Adaptive	Selective
acq	1569	28.6	696	94.43	94.55	95.55	94.43	94.53
crude	253	4.6	212	87.78	90.26	94.07	93.28	92.37
earn	2840	51.8	1083	74.15	74.51	97.16	97.11	82.90
grain	41	0.7	10	88.89	88.89	95.23	95.23	95.23
interest	190	3.5	81	75.38	79.10	85.18	84.47	86.27
money-fx	206	3.8	87	78.15	79.74	81.87	80.65	81.01
ship	108	2.0	36	70.18	70.18	82.86	82.26	81.01
trade	251	4.6	75	87.41	88.41	95.42	95.42	90.90
average	5485	100	2189	**82.046**	**83.205**	**90.793**	**90.356**	**87.926**

the -1 class as their graphs intersect at some positive point. This demonstrates that only using interpolation-based sampling like in-class sampling or SMOTE, in some case one cannot recover the true distribution of small class data and still cannot make a bias decision in favor of the small class.

5.2 Effectiveness of the Algorithms

We evaluate the ability of the proposed algorithms to learn imbalanced data in various domains. The base learner in our experiments was Support Vector Machines [15], due to its high performance on a vast number of domains. However, the approaches are meant to be general, not bound to any specific classification method. We showed the performance with *F measure*, defined as: $F\ measure = 2* \frac{pr*rc}{pr+rc}$ where pr and rc respectively are the precision and recall of the learner on the small class. We deliberately choose domains in which data potentially has manifold structures: Reuters-21578[1] and 20 newsgroups[2].

For the two databases, we carried out standard preprocessing as follows. First, we filtered out multiple label documents. Non-letter characters, short words of less than three characters and stop words were removed. We then applied Porter's stemming[3] to the remaining words. Too infrequent words were removed. These steps left Reuters-21578 data with 4172 terms, and 20 newsgroups with 17835 terms. For Reuters-21578, in the end, we chose only eight categories, which gave us large enough number of documents for the experiment. Train-test splitting was recommended by the data sources. Finally, documents were represented using TFIDF.

We used SVMs from the LIBSVM[4] package. We evaluated the effectiveness of the proposed algorithms by comparing their performance against SVMs itself and SMOTE on top of SVMs. Table 1 and 2 show the statistics of the data sets (number of the small class training examples, its percentage and number of the small class testing examples) and the results of each algorithm in a column, namely the plain SVMs, followed by the SMOTE, Monolithic, Adaptive and Selective algorithms. The last rows contain total numbers for statistics and average results of all data sets for algorithms.

Parameters for the algorithms were chosen as follows: For SVMs on text classification, we chose linear kernel. Regularization parameter C was selected from the highest F measure using a cross-validation, C=10 for Reuters-21578 and C=5 for 20 newsgroups. The other algorithms, i.e. SMOTE and our sampling algorithms used the same SVMs parameters. Number of nearest neighbors k=5 for all experiments. For the sampling degrees (in-class and out-class sampling) of our approaches and of SMOTE, as noted previously, they are free parameters; we just run with all parameters and select the highest ones.

For the Reuters-21578 data in Table 1, on average, SMOTE gives a little higher F-measure than plain SVMs (1.159% higher). However, all of the proposed algorithms give much higher results than SVMs (8.747% for Monolithic, 8.310% for Adaptive and 5.880% for Selective). Moreover, our algorithms also show a significant improvement over SMOTE (7.588%, 7.151% and 4.721%). This experiment confirms the merit of proposed sampling algorithms. This also confirms our observation about the drawbacks of interpolation-based samplings like SMOTE and proves that out-class sampling strategy is necessary.

Results for the 20 newsgroups data are shown in Table 2. We can see that, in most cases, our proposed algorithms give some improvement over plain SVMs, and a slight improvement over SMOTE. On average, the improvements of those algorithms over SVMs are: SMOTE: 4.083%, Monolithic: 5.049%, Adaptive: 4.611% and Selective: 4.286%. One may conclude that in these data sets, it is likely that out-class sampling is not crucial, that only in-class sampling would be enough. However, using out-class sampling does not damage performance as we can see that the approaches

[1] http://www.daviddlewis.com/resources/ t̃estcollections/reuters21578/

[2] http://people.csail.mit.edu/jrennie/20Newsgroups/

[3] http://www.tartarus.org/martin/PorterStemmer/

[4] http://www.csie.ntu.edu.tw/ cjlin/libsvm/

Table 2. Statistics and F-measure results on 20 newsgroup data.

class	#train	%train	#test	SVMs	SMOTE	Monolithic	Adaptive	Selective
atheism	480	4.3	319	64.43	72.04	72.04	69.85	70.61
graphic	584	5.2	389	68.91	72.14	72.35	71.89	72.09
ms-windows	572	5.1	394	55.41	61.59	64.07	64.06	64.97
pc.hardware	590	5.2	392	61.63	65.07	65.07	65.43	66.07
mac.hardware	578	5.1	385	68.98	72.23	72.66	72.84	72.27
windows.x	593	5.3	392	70.74	74.66	74.93	74.93	73.33
forsale	585	5.2	390	75.77	79.44	81.45	81.17	80.69
autos	594	5.3	395	80.47	82.65	82.65	82.53	82.42
motorcycles	598	5.3	398	87.99	88.80	88.80	88.00	88.83
baseball	597	5.3	397	85.47	86.59	87.86	87.60	86.84
hockey	600	5.3	399	94.22	94.53	94.59	94.42	94.54
crypt	595	5.3	396	86.62	88.80	88.80	87.76	88.62
electronics	591	5.2	393	56.36	65.28	65.28	64.19	65.71
med	594	5.3	396	76.52	80.80	83.81	83.60	81.10
space	593	5.3	394	83.76	86.58	86.58	86.16	86.78
christian	598	5.3	398	77.47	79.62	82.10	82.10	81.36
guns	545	4.8	364	70.83	73.48	74.73	74.27	73.93
mideast	564	5.0	376	78.00	83.13	83.67	83.04	79.62
politics	465	4.1	310	56.32	61.19	64.31	63.49	62.00
religion	377	3.3	251	40.36	53.30	55.48	55.15	54.19
average	11293	100	7528	**72.013**	**76.096**	**77.062**	**76.624**	**76.299**

relying on it still give higher F-measures.

One can see that the thee algorithms perform differently on different data sets. None of them gives the highest F-measure. Therefore, the three algorithms are options for user to choose from. Comparing to the original SVMs and SMOTE, we can see that the three algorithms can utilize the out-class sampling strategy appropriately.

5.3 The merit of out-class sampling

We claimed the deficiency of in-class sampling and SMOTE that they may shrink down the data distribution in term of convex hull and sample variance. The above results show the merit of combining both in-class sampling and out-class sampling in comparison to SMOTE. In order to illustrate the merit of out-class over in-class sampling that out-class sampling also contribute to the performance, we take an example of runs on *interest* and *crude* categories in Reuters-21578 with different parameters as in Table 3 and 4. The examples show that only in-class sampling is insufficient and adding out-class sampling is beneficial. In fact, in both categories, out-class sampling is the main contribution to the improvements.

In summary, we have compared our proposed algorithms with plain SVMs and a sampling algorithm of SMOTE. It was shown that our proposed algorithms give improvements consistently over the plain SVMs. The algorithms can also give significantly higher results than SMOTE. These experiments confirm our claim for the drawbacks of in-class sampling and SMOTE. They also show the merit of manifold assumption.

6 Conclusion and Discussion

In this work, we used the flexible notion of manifold to represent small class data distribution. Relying on the manifold leaning methods, we developed sampling strategies and then combined into three algorithms. Evaluated on text domain, our algorithms showed significant improvements in their ability to learn the small class. It is confirmed that the notion of manifold is flexible enough for data distribution of the small class and is beneficial when data lies on or near a manifold. It is also confirmed that the drawbacks of interpolation-based sampling like in-class sampling and SMOTE are among the effects of imbalanced data. This is an insight into the imbalanced data problem. The proposed algorithms use the out-class sampling strategy appropriately to overcome the drawbacks of interpolation-based samplings.

This work inherits the limitations of manifold learning methods, not suitable for the cases that manifold learning fails to model the data distribution. To go beyond the current work, we think of using more generative models integrating with manifold learning algorithms. We think it is important to investigate different effects of each imbalanced data problem and design more appropriate sampling strategies for each problem accordingly.

Table 3. *interest* **with different in-class and out-class sampling degrees.**

in-class	out-class sampling degree					
	0	1	2	3	4	5
0	75.38	84.56	81.82	85.19	84.66	83.64
1	77.27	84.35	81.82	85.19	84.66	
2	79.10	85.14	83.12	85.19		
3	75.38	84.35	81.82			
4	79.10	83.78				
5	78.20					

Table 4. *crude* **with different in-class and out-class sampling degrees.**

in-class	out-class sampling degree					
	0	1	2	3	4	5
0	87.78	92.31	94.07	92.97	90.77	89.73
1	89.78	92.31	94.07	92.97	90.77	
2	90.27	92.31	94.07	92.97		
3	90.27	92.31	94.07			
4	89.87	92.31				
5	89.87					

In this work, we infact do not learn the manifold of the small class explicitly. Instead, it is left as a metaphor to design suitable algorithms. It is not possible to expect the small class data to be densely sampled for manifold learning algorithms to work well. However, only some part of the small class densely connected would suffice to apply our algorithms. If the small class is not densely sampled anywhere, it would be difficult for any method to detect any regularity to learn the small class. The benefit of assuming manifold lie in the fact that manifold is a flexible notion; it can characterize complicated distributions as we are likely to expect in imbalanced data problem.

References

[1] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *ECML*, pages 39–50, 2004.

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sapling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[3] D. L. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proc. Natl. Acad. Sci. USA*, 100(10):5591–5596, 2003.

[4] C. Elkan. The foundations of cost-sensitive learning. In *Seventeenth International Joint Conference on Artificial Intelligence (IJCAI'01)*, pages 973–978, 2001.

[5] T. Fawcett and F. Provost. Combining data mining and machine learning for effective user profiling. In Simoudis, Han, and Fayyad, editors, *The Second International Conference on Knowledge Discovery and Data Mining*, pages 8–13. AAAI Press, 1996.

[6] T. Ha and H. Bunke. Off-line handwritten numeral recognition by perturbation method. *IEEE transactions PAMI*, 19(5):535–539, May 1997.

[7] N. Japkowicz. The class imbalance problems: Significance and strategies. In *Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000)*, volume 1, pages 111–117, 2000.

[8] T. Jo and N. Japkowicz. Class imbalances and small disjunts. *SIGKDD Explorations*, 6(1), June 2004.

[9] M. Kubat, R. C. Holte, and S. Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30(2-3):195–215, 1998.

[10] A. Lazarevic, L. Ertoz, A. Ozgur, J. Srivastava, and V. Kumar. ”evaluation of outlier detection schemes for detecting network intrusions”. In *Third SIAM International Conference on Data Mining*, May 2003.

[11] M. Maloof, P. Langlay, and R. Nevatia. Generalizing over aspect and location for rooftop location. In *IEEE Workshop on Applications of Computer Vision*, 1998.

[12] S. H. Muggleton, C. H. Bryant, and A. Srinivasan. Measuring performance when positives are rare: Relative advantage versus predictive accuracy — a biological case-study. In *European Conference on Machine Learning*, pages 300–312, 2000.

[13] A. Nickerson, N. Japkowicz, and E. Milios. Using unsupervised learning to guide re-sampling in imbalanced data sets. In *Eighth International Workshop on AI and Statitsics*, pages 261–265, 2001.

[14] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–6, Dec 2000.

[15] B. Schlkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[16] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–23, Dec 2000.

[17] S. Tsumoto. The common medical data sets to compare and evaluate kdd methods. *Journal of Japanese Society for Artificial Intelligence*, 15(5):751–758, 2000.

[18] G. Wu and E. Chang. Adaptive feature-space conformal transformation for imbalanced-data learning. In *Proceedings of Twentieth International Conference on Machine Learning*, pages 816–823, 2003.