SROT: Sparse representation-based over-sampling technique for classification of imbalanced dataset

Xionggao Zou\textsuperscript{1,2}, Yueping Feng\textsuperscript{1}, Huiying Li\textsuperscript{1,2}, Shuyu Jiang\textsuperscript{3}

\textsuperscript{1}Computer science and technology department, Jilin University, 130012 ChangChun, Jilin province, China
\textsuperscript{2}Symbol Computation and Knowledge Engineer of Ministry of Education, Jilin University, ChangChun, Jilin province, China
\textsuperscript{3}College of software, Jilin University, 130012 ChangChun, Jilin province, China

Abstract. As one of the most popular research fields in machine learning, the research on imbalanced dataset receives more and more attentions in recent years. The imbalanced problem usually occurs in when minority classes have extremely fewer samples than the others. Traditional classification algorithms have not taken the distribution of dataset into consideration, thus they fail to deal with the problem of class-imbalanced learning, and the performance of classification tends to be dominated by the majority class. SMOTE is one of the most effective over-sampling methods processing this problem, which changes the distribution of training sets by increasing the size of minority class. However, SMOTE would easily result in over-fitting on account of too many repetitive data samples. According to this issue, this paper proposes an improved method based on sparse representation theory and over-sampling technique, named SROT (Sparse Representation-based Over-sampling Technique). The SROT uses a sparse dictionary to create synthetic samples directly for solving the imbalanced problem. The experiments are performed on 10 UCI datasets using C4.5 as the learning algorithm. The experimental results show that compared our algorithm with Random Over-sampling techniques, SMOTE and other methods, SROT can achieve better performance on AUC value.

1 Introduction
In real life, samples of datasets are usually imbalanced, meaning that some classes have fewer samples than others. Most of existing data is imbalanced, while specific data satisfying balance requirements is difficult to acquire, and is, therefore, very scarce. This situation is very common in network intrusion, medical diagnosis, text classification, and other practical applications \cite{1}. Many kinds of situations can still be converted to binary classification problems, even though most of datasets have multi-class attributes, so this work focuses on binary classification problems.

At present, effective solutions of this problem can take three types of strategies: data pre-processing, algorithm and prediction post-processing \cite{2}. Among these three strategies, the algorithm based on data is more popular because of its independence and adaptability. The strategy based on data side is mainly to solve the imbalance problem by changing the size of such imbalance training datasets under the four criteria proposed by Breiman \cite{3}, namely, to resample the datasets to enforce the majority and minority classes back to balance. This paper is based on the Random Over-sampling technique. The Random over-sampling technique copies majority class samples into the minority class in order to increase it. The SMOTE (Synthetic Minority Over-Sampling Technique) is the most
representative over-sampling technique, proposed by Chawla in 2002 [4]. Besides, the SMOTE contains many other modified algorithms [5], such as the Borderline-SMOTE proposed by Han [6], SMOTE-TL [7] and so on. The Random over-sampling technique can improve the ability of classifiers of unbalanced data, because it can reduce the imbalance degree of the sample space. However, the Random sampling simply copies the original data as a large number of repeated samples, which makes the classifier decision area too small. This eventually leads to the overfitting phenomenon, thus seriously reduces the generalization performance of the algorithm. So we propose an improved algorithm to solve this problem: SROT (Sparse Representation-based Over-sampling Technique) in this paper. The work of this paper is inspired by the widely used SMOTE algorithm and sparse representation theory. First, we use minority class samples to construct sparse dictionary for getting algorithms. The experimental results show that our algorithm effectively improve the ability of the classifier to distinguish imbalanced data sets.

2 Method

This paper puts forward an improved over-sampling techniques based on sparse representation: SROT. Our research work is based on the framework of Compressive Sensing (CS). This section describes the relevant theory before introducing our new algorithms.

2.1 Compressive Sensing and Sparse representation

Compressed Sensing (also known as Compressive Sensing, Compressive Sampling, or Sparse Sampling) is a signal processing technique for efficient acquisition and reconstruction of a signal, by finding solutions of underdetermined linear systems [8]. Nyquist–Shannon sampling theorem [9] shows that if a function x(t) contains no frequencies higher than B hertz, it is completely determined by providing its values at a series of points spaced 1/(2B) seconds apart. This theorem has been used in digital signal processing in past decades. However, Compressive Sensing theorem states that the signal bandwidth is not an essential requirement for sampling, and the signal sampling rate only depends on the sparse signal and incoherence of the sampling system [8]. This theory allows achieving signal compression and sampling simultaneously, and is used in academia and industry frequently.

The core problem of Compressive Sensing is the design of a sparse dictionary and measurement matrix, and construction of a signal reconstruction algorithm, which is also called sparse representation.

Stacking the measurements into the vector R as y ∈ R^M, and let A ∈ R^{M×N} (M ≪ N) a set of basis vectors. Our goal is to use linear simultaneous equations y = Ax to recover x from y. However, the underdetermined system of equations is ill-conditioned, as the number of unknowns is larger than the number of equations, so that there is no solution of the underdetermined system of equations.

Therefore, if we make x sparse, possibly meaning that ∥x∥_0 (the L_0-norm of x) is as small as possible, then the number of unknowns will be decreased significantly, which makes signal reconstruction possible. Sparse representation is shown in Figure 1.

Then, we can obtain the following optimization target as follows:

\[
\left( f_0 \right): \hat{x}_0 = \text{arg min} \|x\|_0, s.t. Ax = y
\]  

(1)

In 2004, Donoho and Elad proved that the solution of (1) is unique [9]. Still, the L_0-minimum problem is a non-convex optimization problem, which is NP-hard, because no feasible solution can be obtained in polynomial time. In 2006, Tao and Donoho proved that the L_1-normal form can substitute the L_0-normal form based on the RIP condition [10]. Both forms have the same sparse solution, but the framework of Compressive Sensing (CS) becomes a convex optimization problem with solutions obtained in polynomial time based on the L_1-normal form. Its optimization target is as follows:

\[
\left( f_1 \right): \hat{x}_1 = \text{arg min} \|x\|_1, s.t. Ax = y
\]

(2)
Using this, the framework of Compressive Sensing (CS) has been formed originally. In reality, we usually use (3) instead of (2) to take noise into account.

\[
\{t_i^*\}: \hat{x}_i = \arg\min_x \|x\|_1, \text{s.t.} \|Ax - y\|_2 \leq \varepsilon
\]  

where \(A\) represents the sparse dictionary, and \(x\) is called the sparse solution.

The goal of the reconstruction algorithm is to find the solution \(x\), where the core of the whole problem is the sparse representation of \(y\).

![Figure 1: Schematics of sparse representation](image)

2.2 SROT

The SMOTE is an interpolation algorithm based on k-nearest-neighbour. The algorithm cannot change the distribution of data points. When the distribution of the minority class is too sparse, the SMOTE becomes limited in use. Therefore, we attempt to improve over-sampling technique with sparse representation. First, we construct a sparse dictionary, and then SROT uses the sparse dictionary to produce new samples.

The construction of a sparse dictionary includes human construction and training learning. The former contains the isotropic Gabor dictionary \([11]\), anisotropic Refinement-Gaussian dictionary \([12]\) etc. The latter contains the dictionary learning algorithm K-SVD \([13]\). So, we use training samples directly to construct the sparse dictionary in this paper.

Our method provides a training set, and then all minority class samples \(S_{\text{min}} \in \mathbb{R}^{m \times n}\) are detached from that training set. Let \(m\) represent the number of samples, and \(n\) be the dimension of samples. For each current sample point \(x_i \in S_{\text{min}}\), the method uses the rest of minority class samples (all points in \(S_{\text{min}}\) except for \(x_i\)) to construct the sparse dictionary \(D \in \mathbb{R}^{n \times (m-1)}\). In our algorithm, a sample point is represented by a column vector in sparse representation, so the transposition relations of \(D\) and \(S_{\text{min}}\), as well as \(x\) and \(x_i\), are significant.

Next, the paper normalizes every sample point, and calculates their L2-norm as follows.

\[
y_{i,j} = \frac{y_{i,j}}{\sqrt{\sum_j y_{i,j}^2}}, i = 1,2,\ldots,m-1, j = 1,2,\ldots,n
\]  

where \(y_i\) is a sample point of the sparse dictionary \(D\).

Next, we obtain the sparse solution \(w\) of \(x_i\), the following formula is obtained:

\[
x_i = Dw + \varepsilon
\]  

where \(\varepsilon\) represents noise, but \(\varepsilon\) can be ignored due to little differences between the samples of \(D\) and \(x_i\). Next, we modify \(w\) by adding Gaussian noise to the nonzero term of \(w\). The process is described as follows.

\[
w_i = \text{sgn}(w_i) \cdot \left(\text{abs}(w_i) + \text{normrnd}(0,\text{sigma})\right),
\]
i = 1, 2, ..., k  \tag{6}

where k is the number of the nonzero term of w, normrnd(0,sigma) is used to produce Gaussian random numbers with mean 0 and standard deviation sigma. The value of sigma is related to the value of w. The agitation is stronger, the bigger w. Therefore, the following formula is assumed:

\[ \text{sigma} = \text{abs}(\beta w_i) \]

where \( \beta \) is a parameter, which is generally set as \( \beta \in [0.1, 0.8] \) for better experimental results. If the sigma is too small, we obtain meaningless agitation. So, the value of sigma should be 1 at least.

In addition, we can obtain better results if we modify a part of the nonzero term of w instead of the whole nonzero term. \( \alpha \) is used to represent this ratio, \( \alpha \in [0.6, 1] \). Finally, we use some samples of the minority class to create the sparse dictionary and obtain the sparse representation of \( x_i \), considering that the time complexity of the SROT increases when expanding the sparse dictionary. The samples ratio is controlled by \( \gamma \). The overall procedure of the SROT is described below.

**Algorithm**: SROT(T,N,alpha,beta,gamma)

Input: Number of minority class samples T; Amount of Sparse-SMOTE N%; alpha, beta, gamma are the controllable parameters.

Output: Syn_Samp[][], synthetic minority class samples

1. if N<100
2. then Randomize the T minority class samples
3. \( T=(N/100)*T \)
4. N=100
5. endif
6. N=(int)N/100 (* The amount of Sparse-SMOTE is assumed to be in integral multiples of 100.*)
7. Data[][]: array for original minority class samples
8. newindex: keeps a count of number of synthetic samples generated, initialized to 0
9. for i→1 to T
10. Point_i is the i-th minority class samples
11. Choose a group of random number between 1 and T, except for I, call it Index. Size of Index is gamma*T
12. Dic[][] is the samples of minority class samples Data[][] whose indices is in Index. (* Obtain the synthetic samples by Compressed Sensing.*)
13. Syn_Samp[newindex]=Get_Sparse(Dic,Point_i,N)
14. newindex++
15. endfor

Get_Sparse(D,x,N) (*Function to generate the synthetic samples.*)

16. D is the dictionary, x is current sample
17. Normalize every column of D and set L2-norm to 1
18. Solve L1-minimization problem by the equation
\[ \hat{w} = \arg \min w_i, \text{s.t.} Dw - x_i \leq \varepsilon \], and get \( \hat{w} \)
19. nonzero_index is a group of the subscript of nonzero items in \( \hat{w} \)
20. nonzero_num is the number of nonzero items in \( \hat{w} \)
21. for r→1 to N
22. \( w_1 = \hat{w} \)
23. \( k=\alpha*\text{nonzero_num} \)
24. Choose a group of random number between 1 and nonzero_num, call it array. Size of array is k
25. for attr→1 to k
26. \( i=\text{nonzero_index[array[attr]]} \)
27. \( \text{sigma} = \text{sigma}[\hat{w}_i] \)
28. if sigma<1
29. then sigma=1
30. endif
31. \( w_1(i) = \text{sgn}(\hat{w}(i)) * \text{abs}(\hat{w}(i)) + \text{normrnd}(0, \text{sigma}); \)
32. endfor
33. Compute \( D^*w_1 \), and add its transpose into synthetic array.
34. endfor
35. Return synthetic array

End of SROT

Figure 2 shows the composition of the SROT. The data distribution is the same as above. This algorithm calculates the solution of every sample point, and adds random Gaussian noise to sparse solutions. Then, new sample data points are composited using the sparse dictionary. New synthetic sample points are marked using pentagrams in (b). The final synthetic result is shown in (c).

Figure 2: (a) Original data distribution. (b) Current point \( x_i \), and nonzero point of sparse solution. (c) Final synthetic result SROT

Figure 3: (a) Original data distribution. (b) Current point \( x_i \) and its 5-nearest-neighbors data points. (c) Synthetic result obtained by interpolating point 2 to current point. (d) Final synthetic result of SMOTE
Compared with Figure 3 result of the SMOTE, we see that the sample points generated by the SROT are distributed more randomly, even though a few sample points are out of the distribution area of the original samples, being outlier points. It means that SROT can make data more uniform.

3 Result and analysis

3.1 Experimental design
We perform experiments on 10 UCI datasets, and combine our new method on the original dataset with another two common over-sampling techniques: the Random Over-Sampling and SMOTE. After relevant data synthesizing, we use the C4.5 decision tree classifier, AUC (Area Under the ROC curve) as evaluation parameter.

The information about our experimental datasets is shown in Table 1.

| Data Set   | Scale | Property | Objective Value | Minority Class /Majority Class | Imbalanced Degree |
|------------|-------|----------|-----------------|---------------------------------|-------------------|
| Abalone    | 4177  | 1N,7C    | Ring=7          | 391/3786                        | 9.7               |
| Balance    | 625   | 4C       | Balance         | 49/576                          | 11.8              |
| Ionosphere | 351   | 33C      | bad             | 126/225                         | 1.8               |
| Letter     | 20000 | 16C      | A               | 789/19211                       | 24.3              |
| Mf-morph   | 2000  | 6C       | Class 10        | 200/1800                        | 9.0               |
| Mf-zernike | 2000  | 47C      | Class 10        | 200/1800                        | 9.0               |
| Pima       | 768   | 8C       | Class 1         | 268/500                         | 1.9               |
| Satimage   | 6435  | 36C      | Class 4         | 626/5809                        | 9.3               |
| Vehicle    | 846   | 18C      | opel            | 212/634                         | 3.0               |
| Wpbc       | 198   | 33C      | recur           | 47/151                          | 3.2               |

In Table 1, the value of the sample attribute includes Nominal and Continuous marked by N and C, respectively. We cannot obtain nominal attributes by calculations, so that a vectorization method is used. Vectorization refers to using a k-dimensional vector instead of the nominal attribute containing k kinds of values. For example, if any nominal attribute has three values A, B and C, then we use a vector such as (1,0,0) instead of the three nominal values. After vectorization, the dimension is expanded to k-1 based on the original dimension.

The problem of multi-class classification can be converted to a binary classification problem to determine the optimal value. All samples conforming to the optimization value of this sample are classified as the minority class, while the others are in the majority class. We can obtain the optimization value of every dataset, and process imbalance shown in Table 1. However, some original UCI datasets are distributed regularly in some parts, so we need to break the concentrated samples, and vectorize nominal attributes before experiments. We also hope to restore the original distribution of the data to study data. Therefore, we round the data to be integer after synthesis. Finally, we adopt 10 cross validations to assess the synthesis property. Cross-validation is a model validation technique for assessing how the results of statistical analysis are generalized to an independent dataset. In this paper, all samples are divided into N pieces, and one sample becomes a test set, while the others become training sets. The final assessment results are the N average assessed estimates. N is usually set to 5 or 10, and 10 is used in this paper. We take the average of five experiments using ten cross validations as a model to evaluate the ultimate value to reduce randomness.

Our experiment is conducted in MATLAB based on the Weka machine learning and data mining software. The classifier algorithm adopts the J48 algorithm (C4.5 decision tree) of the Weka with default parameter settings. The parameters of the SROT are set as follows: Alpha=0.9, Beta=0.5 and Gamma=0.2.

3.2 Comparison of classification result
Table 2 and Figure 4 show the AUC values of the four processing methods on ten datasets. The ratio of synthesis is set to the value which makes the dataset balanced. For instance, for Pima dataset whose imbalance degree is 1.9, its ratio of synthesis is set to 100. It can be relatively balanced after sample composition, when the ratio of the minority class to the majority class is 536/500.

It can be seen that four kinds of sampling techniques can enhance classification performance on unbalanced datasets with the classifier compared to the original data processing. But the Random Over-Sampling leads to classifier over-fitting, resulting in limited performance improvement, which is proved in the experiments. On the other hand, the SMOTE can avoid over-fitting, and improve the performance of the imbalance problem using the over-sampling technique and interpolation theory. But performance on datasets using the SMOTE is not always better compared to the Random Over-Sampling. For example, Abalone and Ionosphere datasets show poor performance when using the SMOTE. The experiments are proved that there is no absolutely best algorithm to deal with all imbalanced datasets.

The SROT outperforms the SMOTE on almost half of the datasets, including Abalone, Balance and Mf-zernike. Especially on Balance dataset, the SROT significantly outperforms both SMOTE and Random Over-sampling. However, its performance is less effective for Pima, Satimage, Vehicle and Wphc. In spite of this, the SROT algorithm can improve classification results of the imbalance problem.

In order to verify the effectiveness of our algorithms, we compare our method with other methods for over-sampling. In Figure 5, the average AUC values over all 10 datasets of Table 1 are given for each method. It can be seen that SROT improve the Random Over-Sampling, SMOTE, SMOTE-BL1 and SMOTE-BL2, SMOTE-TL and SMOTE-RSB [16], but do not improve SMOTE-FRST [17]. Nevertheless, Figure 5 show that if we use sparse representation for over-sampling, we still obtain good results.

|                  | None   | RO     | SMOTE  | SROT   |
|------------------|--------|--------|--------|--------|
| Abalone          | 0.500± | 0.674± | 0.726± | 0.817± |
| Balance          | 0.500± | 0.508± | 0.521± | 0.715± |
| Ionosphere       | 0.883± | 0.891± | 0.881± | 0.901± |
| Letter           | 0.987± | 0.974± | 0.985± | 0.990± |
| Mf-morph         | 0.500± | 0.903± | 0.923± | 0.926± |
| Mf-zernike       | 0.107± | 0.717± | 0.776± | 0.796± |
| Pima             | 0.758± | 0.714± | 0.735± | 0.704± |
| Satimage         | 0.761± | 0.771± | 0.806± | 0.767± |
| Vehicle          | 0.729± | 0.710± | 0.753± | 0.755± |
| Wpbc             | 0.623± | 0.590± | 0.667± | 0.637± |
In order to compare the results, we perform a statistical analysis conducted by non-parametric multiple comparison procedures \cite{18} to find better pre-processing algorithms. We use Friedman’s procedure to compute the set of ranks that represent the effectiveness associated with each algorithm. In Table 3 we can observe that our proposals have great ranking.

| Methods     | Ranking |
|-------------|---------|
| SMOTE-FRST  | 3.166   |
| SROT        | 3.278   |
| SMOTE-RSB   | 3.564   |
| SMOTE-TL    | 3.997   |
| SMOTE       | 4.426   |
| SMOTE-BL1   | 5.830   |
| SMOTE-BL2   | 6.090   |
| RO          | 6.749   |

4 Conclusion and future work

The imbalance problem of datasets is one of the hottest areas of research in machine learning, pattern recognition, and other fields. Traditional classification algorithms that do not consider the distribution of the sample tend to reduce the majority class, leading to performance fail sharply. So we put forward
SROT algorithm. According to performance results on ten UCI datasets, The SROT can improve this issue uniformity in spite of the consequences that are likely to cause data overlapping and noise enhancing. Therefore, the SROT can significantly improve the performance of classifiers, and effectively solve the non-equilibrium problem compared to the original dataset and some random sampling techniques.

In the future we want to take this work a step further by applying more and better data cleaning techniques on datasets pre-processed by our methods.

Acknowledgements
This study was supported by Scientific and Technological Development Scheme of Jilin Province under Grant No. 20140520071JH.

References
[1] Chen E, Lin Y, Xiong H, et al. “Exploiting probabilistic topic models to improve text categorization under class imbalance”, Information Processing & Management, 47(2), pp. 202-214, (2011).
[2] Branco P, Torgo L, Ribeiro R P. “A survey of predictive modeling on imbalanced domains”, ACM Computing Surveys (CSUR), 49(2), pp. 31, (2016).
[3] Breiman L, Friedman J, Stone C J, et al. “Classification and regression trees”, CRC press, (1984).
[4] Chawla N V, Bowyer K W, Hall L O, et al. “SMOTE: synthetic minority over-sampling technique,” Journal of artificial intelligence research, pp. 321-357, (2002).
[5] Wang B X, Japkowicz N. “Imbalanced data set learning with synthetic samples//Proc”, IRIS Machine Learning Workshop, pp. 19, (2004).
[6] Han H, Wang W Y, Mao B H. “Borderline-SMOTE: a new over-sampling method in imbalanced datasets learning//Advances in intelligent computing”, Springer Berlin Heidelberg, pp. 878-887, (2005).
[7] Batista G E, Prati R C, Monard M C. “A study of the behavior of several methods for balancing machine learning training data”, ACM Sigkdd Explorations Newsletter, 6(1), 20-29, (2004).
[8] Donoho D L. “Compressed sensing”, IEEE Transactions on Information Theory, 52(4), 1289-1306, (2006).
[9] Jerri A J. “The Shannon sampling theorem—Its various extensions and applications: A tutorial review”, Proceedings of the IEEE, 65(11), 1565-1596, (1977).
[10] Tsaig Y, Donoho D L. “Extensions of compressed sensing”, Signal processing, 86(3), 549-571, (2006).
[11] Bergeaud F, Mallat S. “Matching pursuit of images//Time-Frequency and Time-Scale Analysis, 1994.”, Proceedings of the IEEE-SP International Symposium on, IEEE, 330-333, (1994).
[12] Figueras i Ventura R M, Vanderheyst P, Frossard P. “Low-rate and flexible image coding with redundant representations”, IEEE Transactions on Image Processing, 15(3), 726-739, (2006).
[13] Michal A, Elad M, Alfred B. “K- SVD: An algorithm for designing over-complete dictionaries for sparse representation”, IEEE Transactions on Signal Processing, 54(11), 4311-4322, (2006).
[14] C.Blake, E.Keogh, and C.J. Merz. “UCI repository of machine learning databases”, https://archive.ics.uci.edu/ml/datasets.html, Department of Information and Computer Science, University of California, Irvine, CA.
[15] Swets J A. “Measuring the accuracy of diagnostic systems” Science, 240(4857), 1285-1293, (1988).
[16] Ramentol E, Verbiest N, Bello R, et al. “SMOTE-FRST: a new resampling method using fuzzy rough set theory”, 10th International FLINS conference on uncertainty modelling in knowledge engineering and decision making (to appear), (2012).

[17] Ramentol E, Caballero Y, Bello R, et al. “SMOTE-RSB*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory”, Knowledge and information systems, 33(2), 245-265, (2012).

[18] García S, Fernández A, Luengo J, et al. “A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability”, Soft Computing, 13(10), 959-977, (2009).