Journal of Data Analysis and Information Processing, 2015, 3, 27-33
Published Online August 2015 in SciRes. http://www.scirp.org/journal/jdaip
http://dx.doi.org/10.4236/jdaip.2015.33004

An Improved Algorithm for Imbalanced Data and Small Sample Size Classification

Yong Hu[1*], Dongfa Guo[1], Zengwei Fan[1], Chen Dong[1], Qiuhong Huang[1], Shengkai Xie[1], Guifang Liu[1], Jing Tan[1], Boping Li[1], Qiwei Xie[2]

[1]Analytical Laboratory Beijing Research Institute of Uranium Geology, Beijing, China
[2]Department of Electronics and Information, Toyota Technological Institute, Nagoya, Japan
Email: *yonghu_iacas@163.com

Received 18 June 2015; accepted 5 July 2015; published 8 July 2015

Abstract

Traditional classification algorithms perform not very well on imbalanced data sets and small sample size. To deal with the problem, a novel method is proposed to change the class distribution through adding virtual samples, which are generated by the windowed regression over-sampling (WRO) method. The proposed method WRO not only reflects the additive effects but also reflects the multiplicative effect between samples. A comparative study between the proposed method and other over-sampling methods such as synthetic minority over-sampling technique (SMOTE) and borderline over-sampling (BOS) on UCI datasets and Fourier transform infrared spectroscopy (FTIR) data set is provided. Experimental results show that the WRO method can achieve better performance than other methods.

Keywords

Class Imbalance Learning, Over-Sampling, High-Dimensional Small-Sample Size, Support Vector Machine

1. Introduction

Imbalanced data [1] sets can lead to the traditional data mining algorithms behaving undesirable, which is because the distribution of the data sets is not taken into consideration in the algorithms. Because of the extreme imbalance, a trivial learning algorithm may cause the decision boundary skewed toward the minority class, so the new minority test samples are likely to be misclassified. Various methods for dealing with this problem have been proposed recently. The first type of methods focuses on data processing: removing a number of samples

[*]Corresponding author.

How to cite this paper: Hu, Y., Guo, D.F., Fan, Z.W., Dong, C., Huang, Q.H., Xie, S.K., Liu, G.F., Tan, J., Li, B.P. and Xie, Q.W. (2015) An Improved Algorithm for Imbalanced Data and Small Sample Size Classification. *Journal of Data Analysis and Information Processing*, **3**, 27-33. http://dx.doi.org/10.4236/jdaip.2015.33004

from the majority class (under-sampling) or adding new samples into the minority class (over-sampling). The former methods [2] have drawbacks that they may lead to lose relevant information. The later method [3] is achieved by adding some synthetic samples until the desired class ratios are attained: Chawla *et al.* [3] over-sample the minority class through synthetic minority over-sampling technique (SMOTE) method. Nguyen *et al.* [4] propose borderline over-sampling (BOS) method in which only the minority samples near the borderline are over-sampled. The second type of methods focuses on modifying the existing classification algorithms. For support vector machines (SVM) method, proposals such as using different weighting constants for different classes [5], or adjusting the class boundary based on kernel-alignment ideal [6] are reported. Huang *et al.* [7] present biased minimax probability machine (BMPM) to resolve the imbalanced problem. Furthermore, there are other effective methods such as cost-sensitive learning [8] and one-class learning [9].

In the particular tasks such as face recognition (FR) [10], the number of available training samples is usually much smaller than the dimensionality of the samples pace. Consequently, the biggest challenge that all linear discriminant analysis (LDA)-based approaches have to face is the "small sample size" (SSS) problem. These are often ill-posed problems. There are many ways to address the problem: One option is to apply linear algebra techniques to solve the numerical problem of inverting the singular within class scatter (WCS) matrix. The second option is the feature extraction-based methods, such as the well-known fisher faces method [11]. However, the discarded null space may contain significant discriminatory information, and this will further effect the formation of classifier. The third option is over-sampling method: we can over-sample the training samples so that the number of samples is comparable with the dimensionality of the samples pace, which will make the WCS nonsingular.

We solve the imbalanced problem and SSS problem based on data processing. To deal with the two problems, we propose a windowed regression over-sampling (WRO) method. In this method, the virtual samples are generated according to the difference between adjacent samples. In contrast to SMOTE and BOS methods, the difference is estimated in a local window with the least square regression instead of the whole ones. Moreover, both additive and multiplicative effects between samples are considered in WRO algorithm.

2. Weighting Support Vector Machines for Classification

The objective of the training of SVM is to find the optimal hyperplane that separates the positive and negative classes with a maximum margin [12]. Consider the training set $\{(x_i, y_i)\}, i = 1, \cdots, n$, where x_i is a training sample and $y_i \in (+1, -1)$ is its corresponding true label. To solve the imbalanced datasets, Veropoulos *et al.* [5] suggested using different weighting constants for the minority and majority classes in SVM (Weighting SVM: WSVM):

Minimize:

$$\frac{1}{2}\|\omega\|^2 + C^+ \sum_{\{i|y_i=+1\}} \xi_i + C^- \sum_{\{i|y_i=-1\}} \xi_i \tag{1}$$

Subject to:

$$y_i\left(\omega \times \phi(x_i) + b\right) \geq 1 - \xi_i \quad i = 1, \cdots, n \tag{2}$$

where ω and b are the weight vector and the bias of the hyperplane respectively, ξ_i indicates degree of location violation of the i-th training sample, C^+ and C^- are the different error costs for the minority and majority classes. $K(x_i, x_j) = \phi(x_i)^{\mathrm{T}} \phi(x_j)$ is akernel function that enables to compute dot products in the feature space without knowing the mapping ϕ. In this paper, we use the RBF kernel as follows:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \tag{3}$$

where γ is a width parameter, control the radial scope. There are no guidelines for deciding what the relative ratios of the minority to majority cost factors should be, we empirically set the cost ratio to the inverse of the imbalance ratio and that is what we have used in this paper. However, WSVM is sensitive to the minority samples and obtains stronger cues from the minority samples about the orientation of the plane than from the majority samples. If the minority samples are sparse, as in imbalanced datasets, then the boundary may not have the proper shape in the input space [13].

3. The Proposed Algorithm

To solve the imbalanced problem, an appropriate number of virtual samples are added to the minority class according to the sampling level; to solve the SSS problem, we generate virtual samples so that the size and the dimensionality of training samples are comparable to a certain extent. The basic idea is as follows:

Let $X_{n\times m}$ be a samples matrix whose rows and columns correspond to samples and variables respectively. Denote the n samples as $x_1, x_2, \cdots, x_n$, we produce more virtual samples in the dense region and less in the sparse region: calculating the mean of the samples in the category and denoting it as *x*, then computing the distance between the mean value and each sample $d_i = \|x_i - \overline{x}\|_2 \; i = 1,\cdots,n$ and obtaining the normalized weight vector $W(w_1, w_2, \cdots, w_n)$ for each sample as follows:

$$w_i = \frac{\widehat{w_i}}{\sum_i \widehat{w_i}}, \quad \widehat{w_i} = \frac{1}{d_i} \tag{4}$$

the weight w_i reflects the *i*-th sample distribution in the training set.

Given the sampling level $p(1,2,\cdots,q)$, we will generate a total of *T* virtual samples; it means generate T_i virtual samples corresponding to each sample x_i:

$$T_i = [T \times w_i], \quad i = 1,\cdots,n \tag{5}$$

where $[\cdot]$ stands for backing to the nearest integer. The details of generating virtual samples are as follows: firstly, for each sample x_i, we compute its x_i nearest neighbors and denote them as $y_{i1}, \cdots, y_{ik}$, then obtain the regression coefficients in a local window:

$$y_{ik}^{w_j} = a_{ik}^{w_j} \times x_{ik}^{w_j} + b_{ik}^{w_j} \tag{6}$$

where w_j is a local window centered at variable j, $y_{ik}^{w_j}$ and $x_{ik}^{w_j}$ are the *j*-th window part of y_{ik} and x_i respectively, $a_{ik}^{w_j}$ and $b_{ik}^{w_j}$ are the regression coefficients in the local window in the least squares sense. With the sliding window that between the sample x_i and its neighbor y_{ik} correspondingly, we can obtain a series of regression coefficients pair a_{ik} and b_{ik} as shown in **Figure 1**, in order to eliminate the noise impact of the regression coefficients, we use Savitzky-Golay filter [14] to smooth the coefficients. Finally, we randomly select a pair of coefficients and interact them with y_{ik} to generate a new sample:

$$x_{\text{new}} = a_{ip} \times y_{iq} + b_{ip}, \;\; \forall \; p,q = 1,\cdots,k \tag{7}$$

The WRO algorithm is therefore summarized as follows:

Input: sample matrix $X_{n\times m}$, window width l, number of generation virtual samples T, number of neighbors k.

Output: virtual sample x_{new}.

1) Compute the number of generation virtual samples T_i for each sample $x_i, i = 1,\cdots,n$ according to Equation (5).

2) Find k nearest neighbors for each sample x_i. Obtain the regression coefficients set a_{ik} and b_{ik} through the given sample x_i and the corresponding k nearest neighbors according to Equation (6).

3) Smooth the regression coefficients set with Savitzky-Golay filter.

4) Generate new samples according to Equation (7).

Many over-sampling algorithms such as SMOTE and BOS only reflect the additive effect between each sample, while our algorithm WRO also reflects multiplicative effect all together from Equation (7) and all of these effect are computed in a local region rather than in a whole region. WRO can enlarge the decision regions and also improve the prediction of the minority class while not sacrificing the accuracy of the whole testing set.

4. Materials

Two data sets from the UCI machine learning repository [15] including Glass (7) and Yeast (5) are used in the experiments. Numbers in parentheses indicate which class is chosen as minority class and all of the remaining classes are combined to create a majority class. We also use 500 Fourier Transform infrared (FTIR) spectra as

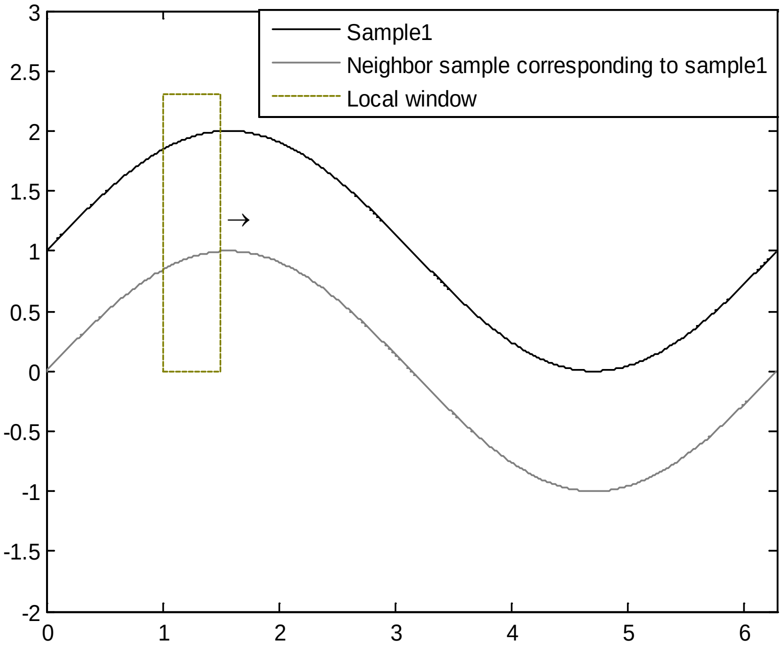

Figure 1. Obtain regression coefficient with the sliding window between samples.

small size data sets. The FTIR spectra in the region 4000 - 650 cm^{-1} have been recorded with a Perkin-Elmer Spectrum GX FTIR spectrometer, equipped with the Universal ATR sampling accessory. The details of UCI data sets and FTIR dataset are provided in **Table 1**. "Imbalance" indicates the ratio between the majority class and the minority class.

5. Experimental Results

The programs are written in house in Matlab Version R2012a and run in a personal computer with a 2.20 GHz Intel Core 2 processor, 4 GB RAM, and a Windows 7 operating system.

5.1. Evaluation Measures

The evaluation measures used for imbalanced samples classification in our experiments are based on the confusion matrix [16]. **Table 2** illustrates a confusion matrix for a two class problem with positive (minority) and negative (majority). With this matrix, our performance measures are expressed: $G\text{-mean} = \sqrt{a^{-} \times a^{+}}$ $F\text{-value} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$, where $a^{-} = \frac{TN}{TN + FP}$, $a^{+} = \frac{TP}{TP + FN}$, $\text{Precision} = \frac{TP}{TP + FP}$, $\text{Recall} = \frac{TP}{TP + FN}$, $G\text{-mean}$ is based on the recalls on both classes. The benefit of selecting this metric is that it can measure how balanced the combination scheme is. If a classifier is highly biased toward one class (such as the majority class), the $G\text{-mean}$ value is low, so it does not depend on the class distribution of the training set. In addition, $F\text{-value}$ combines the recall and precision on the minority class. It measures the overall performance on the minority class. For imbalanced data sets, we apply $G\text{-mean}$ and $F\text{-value}$ as the evaluation measure; for SSS problem, we only apply prediction accuracy as the evaluation measure.

5.2. Experimental Results and Discussions

For imbalanced datasets, we compare the proposed method WRO with WSVM [5] method and some other over-sampling methods including SMOTE and BOS. For SSS problem, we compare the proposed method WRO with standard SVM and PCA feature extraction-based method. The code for SVM and WSVM are taken from the package LIBSVM [17] and the Gaussian RBF kernel is used in the next experiment. We empirically set $l = 3$ for the width of the sliding window and $k = 5$ for the number of neighbors in WRO method. In order to reduce the effect of randomness in the division of data and sampling, each method is run ten times and then the average performance is calculated. Each time consists of: 1) randomly splitting the two classes samples into training and testing sets with the ratio 7.5:2.5; 2) for imbalanced problem, over-sampling the minority class samples on training data with different methods, for SSS problem, over-sampling the two class samples on the training data with different methods; 3) performing 5-fold cross-validation on the over-sampled training data to estimate the optimal parameters C and γ from Equation (3); 4) training SVM classifier; 5) predicting on the test set. Sampling levels are selected according to the imbalance or the relationship between size and dimension of each data set. These over-sampling levels are described in **Table 1**.

Results for Glass are shown in **Figure 2**, we can see that the proposed method WRO achieves a better result in terms of G-mean than that of the other three methods (WSVM, SMOTE, BOS) at almost all the sampling levels, with the growth of oversampling level, the F-value of WRO are comparable with that of the other three methods. For the data set yeast, **Figure 3** shows that three oversampling methods perform well compared to WSVM in terms of G-mean : maybe because of the serious imbalance (Imbalance = 27) for this data set, WSVM is sensitive to the minority samples and obtains stronger cues from the minority samples about the orientation of the plane than from the majority samples, which causes most of the minority samples are misclassified. After over-sampling the minority class, the three oversampling methods improve the results in terms of G-mean , and the F-value evaluation is significantly improved with our method WRO, because the precision evaluation obtained with WRO is better than that of the other three methods.

Figure 4 shows the SSS classification problem, the dimensionality of the sample space is much higher than the

Table 1. Data sets used for the experiment.

Data set	Attributes	No. of data	Imbalance	Sampling levels
Glass	9	214	6	1, 2, 3, 4, 5
Yeast	8	1484	28	1, 4, 8, 12, 16, 20, 24, 27
FTIR	3351	500	1	1, 3, 5, 7, 9

Table 2. Two-class confusion matrix.

	Predicted positive	**Predicted negative**
Actual positive	TP (true positive)	FN (false negative)
Actual negative	FP (false positive)	TN (true negative)

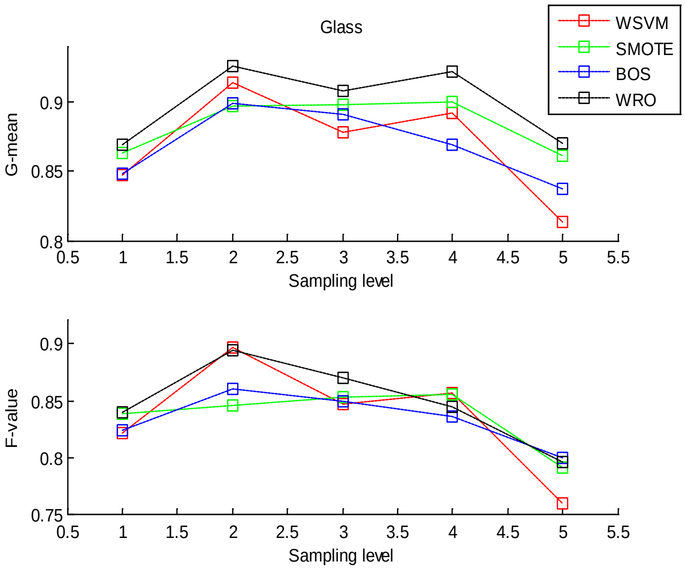

Figure 2. G-mean and F-value performance on the Glass at different sampling level.

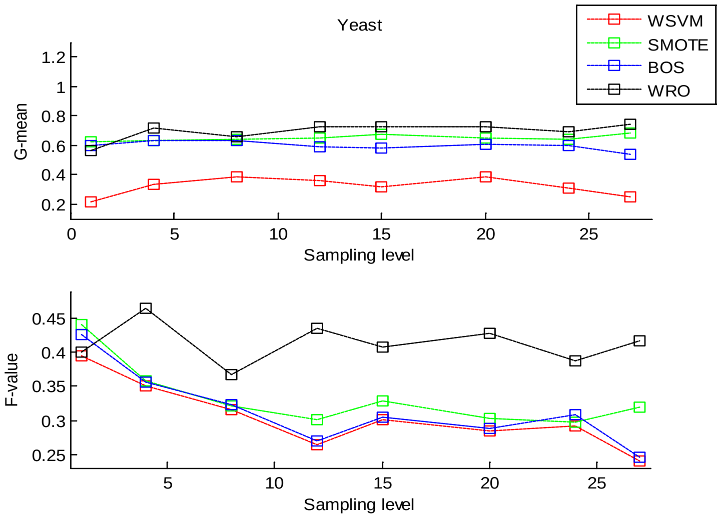

Figure 3. G-mean and F-value performance on the yeast at different sampling levels.

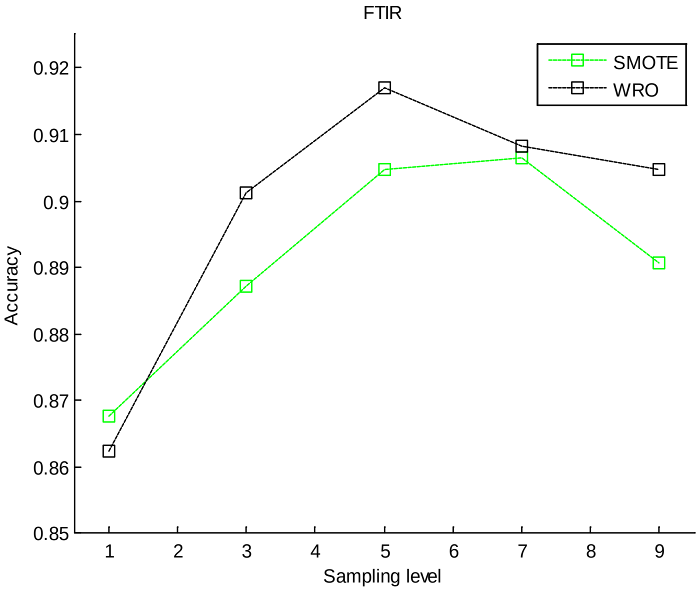

Figure 4. Accuracy on the FTIR at different sampling level.

amount of training samples. Without over-sampling for the training set, the prediction accuracy with SVM is about 86%. After performed with PCA, we used the first ten features, and the prediction accuracy with SVM is about 88% in this case. While the accuracy is improved with SMOTE and WRO methods through an appropriate oversampling level. We can see that the selection of the over-sampling level p impacts on the prediction accuracy of different over-sampling methods, when p is small, we can get better neighbors for the over-sampling process, so the prediction accuracy can be dramatically improved, when p is large enough, more noise is likely to be introduced, so a larger training samples are generated with over-sampling method and less information is lost. Consequently, p is a tradeoff between inducing more noise and losing less information. Nonetheless, our method WRO is comparable with SMOTE method with almost all p values.

6. Conclusion

In this paper, we have addressed the imbalanced data and SSS classification problem. To solve these problems, we propose a new over-sampling method based on windowed regression. Experimental results on two UCI data sets and one FTIR data set demonstrate the efficiency of the proposed algorithm. Of course, there are too many parameters in the algorithm. Meanwhile, the method of solving regression coefficients is in the local window, so the efficiency is not high, and we are going to study all of these.

Acknowledgements

The author thanks for the support by the Core Ability Promotion Project of China National Nuclear Corporation (HXF111-1) and the National Natural Science Foundation of China (61101219).

References

[1] Zheng, Z.H., Wu, X.Y. and Srihari, R. (2004) Feature Selection for Text Categorization on Imbalanced Data. *ACM SIGKDD Explorations Newsletter*, **6**, 80-89. http://dx.doi.org/10.1145/1007730.1007741

[2] Xie, J.G. and Qiu, Z.D. (2007) The Effect of Imbalanced Data Sets on LDA: A Theoretical and Empirical Analysis. *Pattern Recognition*, **40**, 557-662. http://dx.doi.org/10.1016/j.patcog.2006.01.009

[3] Chawla, N. (2003) C4.5 and Imbalanced Data Sets: Investigating the Effect of Sampling Method, Probabilistic Estimate and Decision Tree Structure. *Workshop on Learning from Imbalanced Datasets II*, *ICML*, Washington DC.

[4] Nguyen, H.M., Cooper, E.W. and Kamei, K. (2011) Borderline Over-Sampling for Imbalanced Data Classification. *International Journal of Knowledge Engineering and Soft Data Paradigms*, **3**, 4-21. http://dx.doi.org/10.1504/IJKESDP.2011.039875

[5] Veropoulos, K., Campbell, C. and Cristianini, N. (1999) Controlling the Sensitivity of Support Vector Machines. *Proceedings of the International Joint Conference on AI*, 55-60.

[6] Wu, G. and Chang, E.Y. (2003) Class-Boundary Alignment for Imbalanced Dataset Learning. *Workshop on Learning from Imbalanced Datasets II*, *ICML*, Washington DC.

[7] Huang, K.Z. and Yang, H.Q. (2004) Learning Classifiers from Imbalanced Data Based on Biased Minimax Probability Machine. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **2**, 558-563.

[8] Zhou, Z.H. and Liu, X.Y. (2006) Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem. *IEEE Transactions on Knowledge and Data Engineering*, **18**, 63-77. http://dx.doi.org/10.1109/TKDE.2006.17

[9] Manevitz, L.M. and Yousef, M. (2002) One-Class SVMs for Document Classification. *Journal of Machine Learning Research*, **2**, 139-154.

[10] Samal, A. and Iyengar, P.A. (1992) Automatic Recognition and Analysis of Human Faces and Facial Expressions: A Survey. *Pattern Recognition*, **25**, 65-77. http://dx.doi.org/10.1016/0031-3203(92)90007-6

[11] Belhumeur, P.N., Hespanha, J.P. and Kriegman, D.J. (1997) Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**, 711-720. http://dx.doi.org/10.1109/34.598228

[12] Vapnik, V.N. (2000) The Nature of Statistical Learning Theory. 2nd Edition, Springer, Berlin. http://dx.doi.org/10.1007/978-1-4757-3264-1

[13] Akbani, R., Kwek, S. and Japkowicz, N. (2004) Applying Support Vector Machines to Imbalanced Datasets. Machine Learning: ECML 2004. Springer, Berlin, 39-50. http://dx.doi.org/10.1007/978-3-540-30115-8_7

[14] Luo, J.W., Ying, K. and Bai, J. (2005) Savitzky-Golay Smoothing and Differentiation Filter for Even Number Data. *Signal Processing*, **85**, 1429-1434. http://dx.doi.org/10.1016/j.sigpro.2005.02.002

[15] Asuncion, A. and Jnewman, D. (2007) UCI Machine Learning Repository.

[16] Kubat, M. and Matwin, S. (1997) Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. *Proceedings of the* 14th *International Conference on Machine Learning*, 179-186.

[17] Chang, C.C. and Lin, C.J. (2011) LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology* (*TIST*), **2**, 1-27. http://dx.doi.org/10.1145/1961189.1961199