INCORPORATING COVARIANCE INFORMATION IN ONE CLASS SUPPORT VECTOR CLASSIFICATION

Naimul Mefraz Khan[⋆] *Riadh Ksantini*[†] *Imran Shafiq Ahmad*[†] *Ling Guan*[⋆]

[⋆] Department of Electrical and Computer Engineering, Ryerson University.
[†]School of Computer Science, University of Windsor.

ABSTRACT

Unlike multi-class problems, the low variance directions in the training data are important for one-class classification. However, projecting in these directions before classification will result in loss of important data properties. This paper introduces a Covariance-guided One-Class Support Vector Machine (COSVM) classification method which emphasizes the low variance projectional directions of the training data without compromising any important characteristics. COSVM combines the global information from the covariance matrix of the training data with the local information of Support Vectors. Our proposed method is a convex optimization problem resulting in one global solution, which can be found efficiently with the help of existing numerical methods. The method also keeps the principal structure of the OSVM method intact, and can be implemented easily with the existing OSVM applications. Comparative experimental results with contemporary one-class classifiers on numerous benchmark datasets verify that our method results in significantly better performance.

Index Terms— Covariance, SVM, Outlier Detection.

1. INTRODUCTION

In one-class classification, the objective is to distinguish a particular class of data (*targets*) from other data points (*outliers*) [1]. One-class classification might be necessary when the outlier data is too costly to measure or badly sampled [2].

The two major categories of one-class classifiers are density-based and boundary-based [1]. Density-based methods rely on the estimation of the probability density function (PDF) of the target class [3, 4]. In boundary-based classifiers, the boundary points around the target class are used to classify an incoming data point. Usually, the boundary estimation is formed into a convex optimization problem [5, 2, 6].

The problem with the existing one-class classification methods are that none of them consider the full scale of information available for classification. In density-based methods, solely the overall class probability distribution is used, which can be inaccurate, specially in case of small number of training samples [2]. It is reasonable to assume that the boundary data points are more important than the overall class distribution. In other words, the "local information" available through the boundary data points are not given any special treatment in density-based methods. On the other hand, in boundary-based methods, only boundary data points are considered to build the model. These points do not completely represent the overall class. In other words, the boundary-based methods only consider the available local information. The "global information" available through estimated class distribution is not taken into account. Also, unlike multi-class classification problems, the low variance directions of the target class distribution are crucial for one-class classification [7]. Boundary-based methods do not put any special emphasis on these low variance directions. However, finding the optimal number of directions to retain is also not possible because of the bias-variance dilemma [2], which implies that projecting in specific directions before classification can increase the total error due to loss of important data characteristics.

The motivation behind our proposed method is to use the robustness of the boundary-based classifiers while emphasizing the small variance projectional directions. We want to combine the global information of the training data with the local information obtained through boundary-based methods. Generally, the estimated covariance matrix represents the global information. By incorporating the covariance matrix into the minimization problem of the well-known One-class Support Vector Machine (OSVM) method [6], we can emphasize the low variance directions. We call our proposed method the Covariance-guided One-Class Support Vector Machine (COSVM) method. The degree of emphasis on the covariance matrix can be elegantly controlled through one parameter only (details in Section 3). COSVM does not increase the overall computational complexity of the OSVM method and results in a convex optimization problem with one global optimum solution, which can be found efficiently using existing numerical methods. Since COSVM keeps the basic formulation of the OSVM problem unchanged, it can be implemented through existing OSVM packages with minimal coding.

The rest of the paper is organized as follows: In Sec-

E-mail address of first author: n77khan@ee.ryerson.ca

tion 2, we briefly discuss the OSVM method. In Section 3, we formulate our proposed COSVM method and provide a schematic comparison with OSVM. Section 4 provides various experiments on how to properly utilize COSVM for optimal usage. We also provide comparative results with the Gaussian, Parzen [8], k-NN [9], Support Vector Data Description (SVDD) [2] and OSVM [6] classifiers on numerous benchmark datasets. Finally, Section 5 provides conclusive remarks.

2. ONE-CLASS SVM

Let $\mathcal{X} = \{x_i\}_{i=1}^N$ represent the training dataset of N samples. Since real-world data has inherent non-linearity, SVM-based methods try to map the data samples to a higher dimensional feature space $\mathcal{F}$, where linear classification might be achieved. Let, the target class be mapped to a higher dimensional feature class $\mathcal{F} = \{\Phi(x_i)\}_{i=1}^N$ by the function Φ.

One-class SVM (OSVM) tries to find the hyperplane that separates the training data from the origin with maximum margin. It can be modeled by the following optimization problem:

$$\min_{\mathbf{w}\neq 0,\rho} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} - \rho + \frac{1}{vN}\sum_{i=1}^{N}\xi_i,$$
$$s.t. \quad \mathbf{w}^T\Phi(x_i) \geq \rho - \xi_i,\ \xi_i \geq 0 \ \ \forall i = 1,\dots N. \qquad (1)$$

Here, ξ_i are the slack variables to the optimization problem. $v \in (0, 1]$ is the key parameter, which controls the fraction of outliers and fraction of support vectors (SVs) [6]. $\Phi(x) = (f_1^x, f_2^x, \dots, f_N^x) : \mathcal{X} \to \mathcal{F}$ describes the non-linear mapping from the input space to the feature space for the input variable x. In practice, the *kernel trick* [10] is used to calculate the mapping $\mathcal{F}$, where a kernel function $\mathcal{K}$ calculates the inner products of the higher dimensional data samples: $\mathcal{K}(x_i, x_j) =< \Phi(x_i), \Phi(x_j) >, \forall i, j \in \{1, 2, \dots, N\}$. The solutions $\mathbf{w}^*$ and ρ^* form the decision hyperplane of OSVM.

As stated before, OSVM is a boundary-based method, which only considers the boundary data points (SVs) to build a model of the training data distribution. The small variance projectional directions are not provided any special consideration, which can result in better classification performance.

3. THE PROPOSED METHOD

The purpose of our proposed method is to incorporate the global information available through the estimated covariance matrix of our target class. This global information will retain the low variance projectional directions. By incorporating this into the OSVM optimization problem, we can design a classifier that retains both global and local information and, hence, can provide better performance.

The justification for incorporating covariance matrix into the OSVM optimization problem can also be analyzed from the point of view of discriminant analysis. In Kernel Fisher Discriminant Analysis (KFD)[11], the objective is to learn a weight vector that projects the data points onto a direction that maximizes the between-class variance and minimizes the within-class variance. The optimum weight vector $\mathbf{u}$ can be described by:

$$\mathbf{u} = \sum_{i=1}^{N}\lambda_i\Phi(x_i), \qquad (2)$$

where, $\lambda_i, i = 1, \dots, N$ represent the KFD weight vector coefficients.

Now, by solving the OSVM optimization problem, we will show that OSVM tries to estimate the weight vector components similarly. We use Lagrange multipliers to solve the OSVM optimization problem [6]. By introducing the lagrange variables, Problem (1) becomes the following:

$$L(\mathbf{w}, \rho, \xi, \alpha, \beta) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \rho + \frac{1}{vN}\sum_{i=1}^{N}\xi_i$$
$$-\sum_{i=1}^{N}\alpha_i(w^T\Phi(x_i) - \rho + \xi_i) - \sum_{i=1}^{N}\beta_i - \xi_i. \qquad (3)$$

Setting the derivatives to the primal variables to zero, we obtain:

$$\mathbf{w} = \sum_{i=1}^{N}\alpha_i\Phi(x_i), \qquad (4)$$

and

$$\alpha_i = \frac{1}{vN} - \beta_i \leq \frac{1}{vN},\ \ \sum_{i=1}^{N}\alpha_i = 1. \qquad (5)$$

Substituting Equation (4) Equation (5) into Equation (3), we find the dual problem of OSVM:

$$\min_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha} \qquad (6)$$
$$s.t. \quad 0 \leq \alpha_i \leq \frac{1}{vN},\ \ \sum_{i=1}^{N}\alpha_i = 1,$$

where, $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_N)$. $\mathbf{Q}$ is the kernel matrix for the training data i.e.:

$$\mathbf{Q}(i, j) = \mathcal{K}(x_i, x_j), \qquad (7)$$
$$i = 1, \dots, N;\ \ j = 1, \dots, N.$$

Comparing Equation (4) and Equation (2), we can immediately see that the form of the two weight vectors $\mathbf{u}$ and $\mathbf{w}$ are similar. In both cases, we are trying to find a weight vector that spans across all the training data points. The key difference is the way the components λ_i and α_i are being calculated. KFD uses the within-class and between-class scatter

matrices, while OSVM uses support vectors. As described before, both the global information and local information is important for one-class classification. We need to provide special attention to the small variance directions. In KFD, the within-class distance is minimized. The within-class distance is represented through the within-class scatter matrix, which is analogous to the covariance matrix in the one-class case. Intuitively, we can say that incorporating the covariance matrix into the minimization problem of OSVM will result in putting more emphasis on the low variance directions. Hence, we plug the covariance matrix into the OSVM dual problem, and balance the contribution of the kernel matrix $\mathbf{Q}$ and the covariance matrix through our control parameter η. Our proposed Covariance-guided OSVM (COSVM) method can be described by the following optimization problem:

$$\min_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^T(\eta\mathbf{Q} + (1-\eta)\Delta)\boldsymbol{\alpha} \tag{8}$$
$$s.t. \quad 0 \leq \alpha_i \leq \frac{1}{vN}, \ \sum_{i=1}^{N} \alpha_i = 1,$$

where, Δ is the "kernel covariance matrix" and can be defined as follows [11]:

$$\Delta = \mathbf{Q}(I - 1_N)\mathbf{Q}^T, \tag{9}$$

where, $\mathbf{Q}$ is the kernel matrix (Equation (7)), I is the identity matrix and 1_N is a matrix with all entries $\frac{1}{N}$.

Comparing this with Equation (6), we see that we are modifying the objective function of the optimization problem to incorporate kernel covariance matrix Δ. The extent of "contribution" of our kernel matrix $\mathbf{Q}$ and the covariance matrix Δ is controlled by the parameter η, which can take values between 0 and 1. A value of 0 results in ignoring $\mathbf{Q}$ completely, while a value of 1 results in the opposite. For real-world problems, a value in-between will strike the perfect balance between $\mathbf{Q}$ and the small variance directions obtained through Δ. The proposed method still results in a convex optimization problem, since both $\mathbf{Q}$ and Δ are positive definite [12]. As a result, the solution to this optimization can be found efficiently using numerical methods.

Figure (1) shows a schematic comparison of OSVM and COSVM when the optimal parameter value lies in between 0 and 1 ($0 < \eta < 1$). The linear projection direction for OSVM (depicted by dotted arrows) results in huge overlap between the example target and outlier data (circled by dotted boundary). However, due to the extra importance given to the lower variance directions, the hyperplane for COSVM (the solid line) is pulled towards the direction of small variance. As a result, the COSVM projection direction (depicted by solid arrows) results in much less overlap (circled by solid boundary). In this way, the proposed classifier can result in better overall performance by fusing both local and global information.

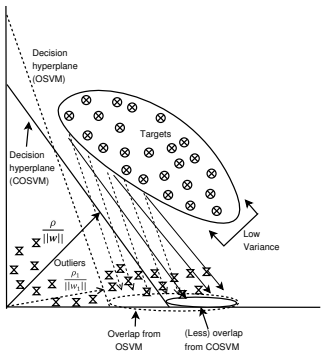


Fig. 1. Schematic comparison of OSVM and COSVM.

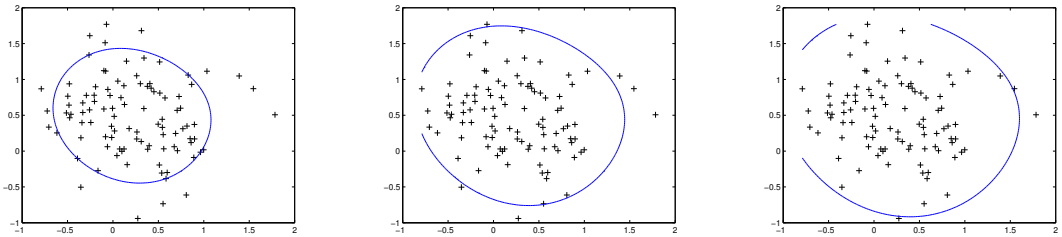
4. EXPERIMENTAL RESULTS

In this section, we provide detailed experimental analysis and results for our proposed method, performed on benchmark real-world datasets and compared against contemporary one-class classifiers. One important step for achieving better classification with COSVM is finding the appropriate value for η. Since in most cases, one-class problems do not have outlier examples, the value of η can't be tuned via cross validation. We use an indirect approach to optimize η. Figure (2) shows the effect of changing the value of η on an artificially generated 2D Gaussian Dataset (radial-basis kernel was used for this experiment. The parameter v for OSVM was fixed to 0.2). We see that the target boundary is being "expanded" as η value decreases. With decreasing η values, more weight is shifted towards the covariance matrix. As a result, the low variance directions are being assigned more importance and the target boundary is being expanded in those directions.

However, we need to use a stopping criterion to find the optimum η value. We use the fraction of outliers as our stopping criterion. The fraction of outliers is determined by calculating what fraction of the training samples are deemed as outliers by the constructed target boundary. We use a predefined lowest fraction of outliers allowed (f_{OL}) as stopping criterion. For a new dataset, we keep slowly decreasing the value of η (starting from 1) and observe the fraction of outliers. When it hits the value of f_{OL}, we stop and use the current η value for that particular dataset. This method of optimization results in superior performance, since we are considering both local and global information.

To depict the performance gain by using COSVM, we compare it with the k-NN, Parzen, SVDD and OSVM classifiers on 8 datasets [1]. We have primarily focused on medical datasets in our experiments, as medical diagnosis is one of the key applications of one-class classification [13]. The datasets were picked carefully to cover varying feature space

[1] Obtained from: `http://prlab.tudelft.nl/users/david-tax`.



(a) $\eta = 1$ (b) $\eta = 0.4$ (c) $\eta = 0$

Fig. 2. COSVM boundaries for different η values.

Dataset	Number of Features	k-NN	Parzen	Gaussian	SVDD	OSVM	COSVM
Haberman's Survival(< 5 years)	3	43.33	44.26	51.36	56.93	*62.85*	**68.37**
Haberman's Survival(> 5 years)	3	62.55	67.97	60.10	68.12	*68.62*	**69.87**
Liver(diseased)	6	60.02	59.9	59.59	60.93	*61.73*	**65.16**
Liver(healthy)	6	50.34	49.69	50.76	62.5	*63.7*	**64.96**
SPECT Images(normal)	44	84.28	*96.45*	93.90	92.32	95.29	**96.79**
SPECT Images(abnormal)	44	19.74	41.29	26.39	57.95	*69.49*	**72.46**
Gene Expression(healthy)	1908	61.5	50	*68.125*	55.5	**72.81**	**72.81**
Gene Expression(tumor)	1908	68.86	50	60.68	67.84	*72.48*	**74.6**

Table 1. Average AUC of each method for the 8 datasets (best method in **bold**, second best *emphasized*). The second column lists the feature dimension for each dataset.

dimensions (second column in Table (1)). The Area Under the Curve (AUC) for the Receiver Operating Characteristic (ROC) curves [14] are presented in our results, which is a common measure of performance for one-class classification [15]. To ensure unbiased results, we average the result over 10 models for each dataset, which were created by removing 10% randomly picked data points from the target class and adding it to the outliers each time. For k-NN, Parzen, Gaussian and SVDD method, we use the popular DDTools toolbox [16]. The parameter *fraction of rejection* [7] was set to 0.2 for these methods. The other necessary parameters are optimized internally within the toolbox. For OSVM and COSVM, we use the SVM-KM toolbox [17]. The parameter v for OSVM was set to 0.2, while f_{OL} for COSVM was set to 0.1. For kernelization in SVDD, OSVM and COSVM, we use the radial basis kernel [10]. The kernel width σ was set to the value for which the fraction of SVs does not decrease any further. This ensures proper scaling of the data [6].

Table (1) contains the average AUC values obtained for the classifiers on each dataset. As we can see, COSVM performs significantly better when compared to other methods in most cases. This strengthens our claim that by emphasizing the small variance directions with the incorporation of the covariance matrix, COSVM can indeed provide improved performance. In general, we see that the performance of k-NN, Gaussian and Parzen classifiers are poor when compared to the SVM-based classifiers (SVDD, OSVM and COSVM). This is because of the limitations inherent in these classifiers. Since k-NN classifies solely based on neighboring points, it is sensitive to outliers [18]. The Gaussian classifier assumes that the underlying distribution is Gaussian, which is not always the case for real datasets. The Parzen classifier is prone to degraded performance in case of high-dimensional data [19], which is clear from the poor results on the *Gene Expression* datasets (1908 features). Among the SVM-based classifiers, COSVM considerably outperforms the other two. As stated before, COSVM combines both local and global information available through the support vectors and the kernel covariance matrix. The balance between these two (obtained through optimizing the control parameter η) results in better performance.

5. CONCLUSION

In this paper, we have proposed the COSVM classification method, which combines both the global information available through the estimated covariance matrix of the training dataset and the local information available through the support vectors. COSVM improves upon the One-Class Support Vector Machine (OSVM) [6] method by emphasizing low variance projectional directions of the training dataset. The proposed method results in a convex optimization problem, which can be solved efficiently with existing numerical methods. Our proposed method does not change the basic formulation of OSVM and can be easily implemented with the existing OSVM applications. Detailed comparative results against five other contemporary one-class classifiers on several benchmark datasets show the superiority of COSVM.

6. REFERENCES

[1] P. Juszczak, *Learning to recognise. A study on one-class classiifcation and active learning*, Ph.D. thesis, Delft University of Technology, Delft, The Netherlands, 2006.

[2] D.M.J. Tax and R.P.W. Duin, "Support Vector Data Description," *Machine Learning*, vol. 54, pp. 45–66, 2004.

[3] L. Tarassenko, P. Hayton, and M. Brady, "Novelty detection for the identification of masses in mammograms," in *IEEE International Conference on Artificial Neural Networks*, Oct. 1995, pp. 442–447.

[4] L. Parra, G. Deco, and S. Miesbach, "Statistical independence and novelty detection with information preserving nonlinear maps," *Neural Computation*, vol. 8, pp. 260–269, 1996.

[5] J. Rosen, "Pattern seperation by convex programming," *Journal of Mathematical Analysis and Applications*, vol. 10, no. 1, pp. 123–134, 1965.

[6] B. Scholkopf, J.C. Platt, J.C. Shawe-Taylor, A.J. Smola, and R.C. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[7] D.M.J. Tax and K.-R. Muller, "Feature Extraction for One-Class Classification," in *Artificial Neural Networks and Neural Information Processing*, 2003, pp. 342–349.

[8] E. Parzen, "On estimation of a probability density function and mode," *Annals of Mathematical Statistics*, vol. 33, pp. 1065–1076, 1962.

[9] S. Harmeling, G. Dornhege, D. Tax, F. Meinecke, , and K.-R. Mueller, "From outliers to prototypes: Ordering data," *Neurocomputing*, vol. 69, no. 13-15, pp. 1608–1618, 2006.

[10] V.N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, 1998.

[11] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.R. Mullers, "Fisher Discriminant Analysis with Kernels," *Neural Networks for Signal Processing*, pp. 41 –48, aug. 1999.

[12] C.A. Michelli, "Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions," *Constructive Approximation*, vol. 2, pp. 11–22, 1986.

[13] L. Guo, "Tumor Detection in MR Images Using One-Class Immune Feature Weighted SVMs," *IEEE Transactions on Magnetics*, vol. 16, no. 10, pp. 3849–3852, 2011.

[14] J.A. Hanley and B.J. McNeil, "A method of comparing the areas under receiver operating characteristic curves derived from the same cases," *Radiology*, vol. 148, no. 3, pp. 839–843, 1983.

[15] G.G. Cabral and A.L. Inacio de Oliveira, "A novel one-class classification method based on feature analysis and prototype reduction," in *IEEE International Conference on System, Man and Cybernetics*, Oct. 2011, pp. 983–988.

[16] D.M.J. Tax, "DDtools, the Data Description Toolbox for Matlab," May 2012, version 1.9.1.

[17] A. Rakotomamonjy, Y. Grandvalet, S. Canu, and V. Guigue, "Svm and kernel methods toolbox," 2007, `http://mloss.org/software/view/33/`.

[18] Y. Jiang and Z.-H. Zhou, "Editing Training Data for k-NN Classifiers with Neural Network Ensemble," in *International Symposium on Neural Networks*, 2004, pp. 356–361.

[19] Y. Muto and Y. Hamamoto, "Improvement of the Parzen classifier in small training sample size situations," *Intelligent Data Analysis*, vol. 5, no. 6, pp. 477–490, 2001.