ONE-CLASS MACHINES BASED ON THE COHERENCE CRITERION

Zineb Noumir, Paul Honeine

Institut Charles Delaunay (CNRS)
Université de technologie de Troyes
10010 Troyes, France

Cédric Richard

Laboratoire H. Fizeau (CNRS)
Université de Nice Sophia-Antipolis
06108 Nice, France

ABSTRACT

The one-class classification problem is often addressed by solving a constrained quadratic optimization problem, in the same spirit as support vector machines. In this paper, we derive a novel one-class classification approach, by investigating an original sparsification criterion. This criterion, known as the coherence criterion, is based on a fundamental quantity that describes the behavior of dictionaries in sparse approximation problems. The proposed framework allows us to derive new theoretical results. We associate the coherence criterion with a one-class classification algorithm by solving a least-squares optimization problem. We also provide an adaptive updating scheme. Experiments are conducted on real datasets and time series, illustrating the relevance of our approach to existing methods in both accuracy and computational efficiency.

Index Terms— support vector machines, machine learning, kernel methods, one-class classification

1. INTRODUCTION

In machine learning, several problems exhibit only a unique class for training. The decision rule consists of identifying if a new instance belongs to the learnt class or to an unknown class. This machine learning problem is the so-called one-class classification problem [1, 2]. It has been applied with success for novelty detection, and extends naturally to tackle multiclass tasks by learning each class separately [3, 4].

Several methods have been developed to solve this problem, the most widely studied being the one-class support vector machines (SVM). It determines a sphere of minimum volume that encloses all (or most of) the training data, by estimating its center and radius. One-class SVM takes advantage of many properties from SVM literature, such as the nonlinear extension by using kernel functions and the sparse solution. The sparsity property states that the center of the sphere explores only a small fraction of the training samples, known as support vectors (SVs). A quadratic programming technique is often applied to solve this problem, i.e., identifying the SVs and estimating the optimal parameters.

This work was supported by ANR-08-SECU-013-02 VigiRes'Eau.

In this paper, we derive a one-class classification machine based on a new sparsification rule, the coherence criterion. This criterion is based on the coherence parameter, a fundamental quantity that describes the behavior of dictionaries in sparse approximation problems. We have recently investigated this sparsification criterion, and applied it with success in nonlinear filtering and online prediction in time series. See [5, 6]. The coherence criterion provides an elegant model reduction criterion with a low computational demanding procedure. This framework allows us to derive new theoretical results, mainly an upper bound on the error of approximating the center by the resulting reduced model. We associate the coherence criterion with a one-class classification algorithm by solving a least-squares optimization problem. We also provide an adaptive updating scheme for incrementing or decrementing the model order.

The rest of the paper is organized as follows. Section 2 outlines the classical one-class SVM. We describe our approach in Section 3, by providing theoretical results and deriving an appropriate one-class classification algorithm. Section 4 provides an experimental study on real datasets and time series.

2. ONE-CLASS SVM

Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ be the set of training samples, et let $\Phi(\cdot)$ be a nonlinear transformation defined by the use of a kernel, $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \Phi(\boldsymbol{x}_i), \Phi(\boldsymbol{x}_j) \rangle$. The one-class SVM finds a sphere, of minimum volume, containing all (or most of) the training samples. This sphere, described by its center $\boldsymbol{c}$ and its radius r, is obtained by solving the constrained optimization problem:

$$\min_{r, \boldsymbol{c}, \boldsymbol{\zeta}} \; r^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \zeta_i$$
$$\text{subject to } \|\Phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 \leq r^2 + \zeta_i \text{ for all } i = 1, 2, \ldots, n$$

In this expression, $\zeta_1, \zeta_2, \ldots, \zeta_n$ is a set of non-negative slack variables and ν a positive parameter that specifies the tradeoff between the sphere volume and the number of outliers, i.e., samples lying outside the sphere. By introducing the Karush-

Kuhn-Tucker optimality conditions, we get

$$\boldsymbol{c} = \sum_{i=1}^{n} \alpha_i \, \Phi(\boldsymbol{x}_i), \tag{1}$$

where the α_i's are the solution to the optimization problem:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}_i) - \sum_{i,j=1}^{n} \alpha_i \alpha_j \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$\text{subject to} \sum_{i=1}^{n} \alpha_i = 1 \text{ and } 0 \leq \alpha_i \leq \frac{1}{\nu n} \text{ for all } i = 1, 2, \ldots, n.$$

It is well known that only a small fraction of the training samples contributes to the model (1). These samples, called support vectors (SVs), correspond to data outside or lying on the boundary of the sphere, and consequently contribute to the definition of the radius r. Thus, any new sample $\boldsymbol{x}$ that satisfies $\|\Phi(\boldsymbol{x}) - \boldsymbol{c}\| > r$ can be considered as an outlier, where the distance is

$$\|\Phi(\boldsymbol{x}) - \boldsymbol{c}\|^2 = \sum_{i,j \in \mathcal{I}} \alpha_i \alpha_j \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) - 2 \sum_{i \in \mathcal{I}} \alpha_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}) + \kappa(\boldsymbol{x}, \boldsymbol{x}),$$

where $\mathcal{I}$ denotes the set of SVs, *i.e.*, training samples with non-zero α_i's. Let $|\mathcal{I}|$ denotes its cardinality.

3. PROPOSED ONE-CLASS METHOD

Let $\boldsymbol{c}_n$ denotes the center of the n samples, namely

$$\boldsymbol{c}_n = \frac{1}{n} \sum_{i=1}^{n} \Phi(\boldsymbol{x}_i). \tag{2}$$

We consider a sparse model of the center with

$$\boldsymbol{c}_{\mathcal{I}} = \sum_{i \in \mathcal{I}} \alpha_i \, \Phi(\boldsymbol{x}_i), \tag{3}$$

where the problem consists of properly identifying the subset $\mathcal{I} \subset \{1, 2, \ldots, n\}$ and estimating the optimal coefficients α_i's. While classical one-class SVM operates a joint optimization, we consider in this paper a separate optimization scheme:

1. Identify the most relevant samples in the expansion (3), by using the coherence parameter in the sparsification rule.

2. Estimate the optimal coefficients, with optimality in the least-squares sense, namely by minimizing $\|\boldsymbol{c}_n - \boldsymbol{c}_{\mathcal{I}}\|^2$.

3.1. Sparsification rule with the coherence criterion

The coherence of a set $\{\Phi(\boldsymbol{x}_i) \mid i \in \mathcal{I}\}$ is defined by the largest absolute value of the off-diagonal entries of the Gram (kernel) matrix, namely a μ-coherent set is[1]

$$\mu = \max_{\substack{i,j \in \mathcal{I} \\ i \neq j}} |\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)|.$$

With the model order being fixed in advance, we consider the set of least coherence as the relevant set in the expansion (3). We have previously applied with success this coherence criterion in nonlinear filtering. See [5, 6]. In order to consider this sparsification rule for the one-class problem, one needs to study the relevance of the sparse model $\boldsymbol{c}_{\mathcal{I}}$ (obtained by the coherence criterion) with respect to the full-order center $\boldsymbol{c}_n$ in (2). The following theorem provides a guarantee that this approximation error is upper bounded.

Theorem 1. *For the sparse solution $\boldsymbol{c}_{\mathcal{I}}$ satisfying the coherence criterion, the approximation error can be upper bounded with*

$$\|\boldsymbol{c}_n - \boldsymbol{c}_{\mathcal{I}}\| \leq \Big(1 - |\mathcal{I}|/n\Big) \sqrt{\max_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}_i) - \mu}. \tag{4}$$

This upper bound takes the form $(1 - |\mathcal{I}|/n)\sqrt{1 - \mu}$ for unit-norm kernels.

Proof. Let $\mathcal{P}_{\mathcal{I}}$ be the projection operator onto the space spanned by the elements $\Phi(\boldsymbol{x}_i)$ for $i \in \mathcal{I}$, thus $\boldsymbol{c}_{\mathcal{I}} = \mathcal{P}_{\mathcal{I}} \boldsymbol{c}_n$. Then, we have

$$\begin{aligned} \|\boldsymbol{c}_n - \boldsymbol{c}_{\mathcal{I}}\| &= \Big\| \frac{1}{n} \sum_{i=1}^{n} (1 - \mathcal{P}_{\mathcal{I}}) \Phi(\boldsymbol{x}_i) \Big\| \\ &\leq \sum_{i=1}^{n} \frac{1}{n} \|\Phi(\boldsymbol{x}_i) - \mathcal{P}_{\mathcal{I}} \Phi(\boldsymbol{x}_i)\| \\ &= \frac{1}{n} \sum_{i \notin \mathcal{I}} \|\Phi(\boldsymbol{x}_i) - \mathcal{P}_{\mathcal{I}} \Phi(\boldsymbol{x}_i)\| \end{aligned}$$

where the inequality is due to the generalized triangular inequality, and the last equality to the fact that $\|\Phi(\boldsymbol{x}_i) - \mathcal{P}_{\mathcal{I}} \Phi(\boldsymbol{x}_i)\| = 0$ for all $\Phi(\boldsymbol{x}_i)$ belonging to the expansion in $\boldsymbol{c}_{\mathcal{I}}$, i.e., for $i \in \mathcal{I}$. Moreover, we have

$$\begin{aligned} \|\Phi(\boldsymbol{x}_i) - \mathcal{P}_{\mathcal{I}} \Phi(\boldsymbol{x}_i)\|^2 &= \|\Phi(\boldsymbol{x}_i)\|^2 - \|\mathcal{P} \Phi(\boldsymbol{x}_i)\|^2 \\ &= \kappa(\boldsymbol{x}_i, \boldsymbol{x}_i) - \max_{\boldsymbol{\gamma}} \frac{\sum_{j \in \mathcal{I}} \gamma_j \kappa(\boldsymbol{x}_j, \boldsymbol{x}_i)}{\| \sum_{j \in \mathcal{I}} \gamma_j \Phi(\boldsymbol{x}_j) \|} \\ &\leq \kappa(\boldsymbol{x}_i, \boldsymbol{x}_i) - \max_{k \in \mathcal{I}} \frac{|\kappa(\boldsymbol{x}_k, \boldsymbol{x}_i)|}{\kappa(\boldsymbol{x}_k, \boldsymbol{x}_k)} \\ &\leq \kappa(\boldsymbol{x}_i, \boldsymbol{x}_i) - \mu \end{aligned}$$

where the first equality is due to the Pythagorian theorem, and the second equality follows the fact that the square norm of the projection of $\Phi(\boldsymbol{x}_i)$ corresponds to the maximum

[1] This definition corresponds to a unit-norm kernel, i.e., $\kappa(\boldsymbol{x}, \boldsymbol{x}) = 1$ for every $\boldsymbol{x}$; otherwise, substitute $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) / \sqrt{\kappa(\boldsymbol{x}_i, \boldsymbol{x}_i) \, \kappa(\boldsymbol{x}_j, \boldsymbol{x}_j)}$ for $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ in the expression.

		optimal one-class SVM	this paper			
			same order as in optimal one-class SVM		order optimal from $\{3,4,\ldots,15\}$	
iris	$n_{\text{train}}\|n_{\text{test}}$	error	error	shared SVs	error	$\|\mathcal{I}\|$
class 0	25|125	1.84	0.49	50%	0.40	3
class 1	25|125	5.28	4.40	40%	1.28	4
class 2	25|125	4.80	4.10	60%	2.88	3
	mean:	**3.97**	**2.99**	50%	**1.52**	3
	time:	*(2.5)*	*(0.6)*		*(0.6)*	
wine						
class 0	29|148	15.48	8.89	80%	8.57	7
class 1	35|142	18.26	22.60	70%	16.51	13
class 2	24|154	14.47	17.70	86%	12.81	13
	mean:	**16.07**	**16.39**	79%	**12.63**	11
	time:	*(22.2)*	*(0.3)*		*(0.3)*	
cancer						
class 0	222|461	2.30	3.03	40%	2.12	9
class 1	119|563	5.21	4.78	46%	3.80	8
	mean:	**4.25**	**3.90**	43%	**2.96**	8
	time:	*(42)*	*(2.4)*		*(2.4)*	

Table 1. Experimental results with the classification error for each one-class classifier, and the mean error, as well as the *total computational time* (in seconds). The ratio of common SVs, with respect to the results obtained from the classical one-class SVM, for each of the proposed methods is given, as well as the mean ratio of common SVs.

scalar product $\langle \Phi(\boldsymbol{x}_i), \varphi \rangle$ over all the unit-norm functions φ, namely $\varphi = \sum_{j\in\mathcal{I}} \gamma_j \Phi(\boldsymbol{x}_j) / \| \sum_{j\in\mathcal{I}} \gamma_j \Phi(\boldsymbol{x}_j) \|$. The first inequality results from a specific distribution of the coefficients, with $\gamma_j = 0$ for all $j \in \mathcal{I}$ except for a single index k with $\gamma_k = \pm 1$, depending on the sign of $\kappa(\boldsymbol{x}_k, \boldsymbol{x}_i)$. The last inequality follows from the coherence criterion. □

3.2. Optimal parameters

The coefficients are estimated by minimizing the approximation error $\|\boldsymbol{c}_n - \boldsymbol{c}_{\mathcal{I}}\|$, namely

$$\boldsymbol{\alpha} = \arg\min_{\alpha_i} \left\| \frac{1}{n} \sum_{i=1}^{n} \Phi(\boldsymbol{x}_i) - \sum_{i\in\mathcal{I}} \alpha_i \Phi(\boldsymbol{x}_i) \right\|^2,$$

where $\boldsymbol{\alpha}$ is a column vector of the optimal coefficients α_k's for $k \in \mathcal{I}$. Taking the derivative of this cost function with respect to each α_k, and setting it to zero, we get

$$\frac{1}{n} \sum_{i=1}^{n} \kappa(\boldsymbol{x}_k, \boldsymbol{x}_i) = \sum_{i\in\mathcal{I}} \alpha_i \kappa(\boldsymbol{x}_k, \boldsymbol{x}_i), \text{ for every } k \in \mathcal{I}.$$

In matrix form, we obtain

$$\boldsymbol{\alpha} = \boldsymbol{K}^{-1} \boldsymbol{\kappa}, \tag{5}$$

where $\boldsymbol{K}$ is the Gram (kernel) matrix, with entries $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for $i, j \in \mathcal{I}$ and $\boldsymbol{\kappa}$ is a column vector with entries $\frac{1}{n}\sum_{i=1}^{n} \kappa(\boldsymbol{x}_k, \boldsymbol{x}_i)$ for $k \in \mathcal{I}$.

An interesting property of the proposed method is that one can easily increment or decrement the model order, without the need to operate from scratch as in SVM. For instance[2], one does not need to inversion the new Gram matrix. Let $\boldsymbol{K}_m^{-1}$ be the Gram matrix defined on the set $\mathcal{I}$ with cardinality m. If one wishes to increment the model order to $m+1$ by injecting the entry $\boldsymbol{x}_k$, then we have

$$\boldsymbol{K}_{m+1} = \begin{bmatrix} \boldsymbol{K}_m & \boldsymbol{b} \\ \boldsymbol{b}^\top & \kappa(\boldsymbol{x}_k, \boldsymbol{x}_k) \end{bmatrix}$$

where $\boldsymbol{b}$ is the vector with entries $\kappa(\boldsymbol{x}_k, \boldsymbol{x}_i)$ for all $i \in \mathcal{I}$. Then, the inverse of $\boldsymbol{K}_{m+1}$ can be computed using the block matrix inversion identity, with

$$\boldsymbol{K}_{m+1}^{-1} = \begin{bmatrix} \boldsymbol{K}_m^{-1} & \boldsymbol{0} \\ \boldsymbol{0}^\top & 0 \end{bmatrix} + \frac{1}{c} \begin{bmatrix} -\boldsymbol{K}_m^{-1}\boldsymbol{b} \\ 1 \end{bmatrix} \begin{bmatrix} -\boldsymbol{b}^\top \boldsymbol{K}_m^{-1} & 1 \end{bmatrix}, \tag{6}$$

where $c = \kappa(\boldsymbol{x}_k, \boldsymbol{x}_k) - \boldsymbol{b}^\top \boldsymbol{K}_m^{-1} \boldsymbol{b}$ and $\boldsymbol{0}$ is a m-length column vector of zeros. The decremental update can be easily derived since, by using the notation

$$\boldsymbol{K}_m^{-1} = \begin{bmatrix} \boldsymbol{Q}_{m-1} & \boldsymbol{q} \\ \boldsymbol{q}^\top & q_0 \end{bmatrix},$$

we obtain from (6) the update $\boldsymbol{K}_{m-1}^{-1} = \boldsymbol{Q}_{m-1} - \boldsymbol{q}\boldsymbol{q}^\top / q_0$.

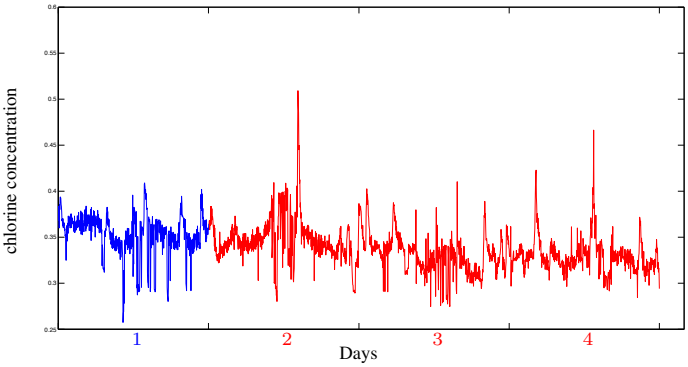

Fig. 1. Time series of the chlorine concentration, with one day for training (blue) and the next 3 days for error estimation (red).

4. EXPERIMENTATION

Multiclass classification

We have tested on three real datasets from the UCI machine learning repository, and well-known in the literature of one-class machines [7]: "iris" with 150 samples in 3 classes and 4 features (only third and fourth features are often investigated), "wine" with 178 samples in 3 classes and 13 features, and "breast cancer" from Wisconsin with 683 samples in 2 classes and 9 features.

[2]Updating expressions for the vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\kappa}$ can be easily derived, and were omitted due to space limitation.

In order to provide a comparative study, we considered the same configuration as in [8]. The Gaussian kernel was applied, with $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/2\sigma^2)$ where σ is the tunable bandwidth parameter. For an ℓ-class classification task, a one-class classifier was constructed for each class, called target class. The target set was randomly partitioned into two subsets, one used for training and the other for the target test set. The whole test set contains the target test set and all the samples from the other classes. The classification error were estimated with the optimal parameters obtained by a ten-fold cross-validation on a grid search over $\nu \in \{2^{-5}; 2^{-4}; \cdots; 2^4; 2^{-1.5}\}$ and $\sigma \in \{2^{-5}; 2^{-4}; \cdots; 2^4; 2^5\}$.

To compare the proposed method to one-class SVM (which is also comparable to [8]), we first considered the same number of SVs as in SVM. Table 1 gives the classification errors for both methods, as well as the ratio of shared SVs. We found that our approach is very competitive with the same model order as SVM, with up to 80% of common SVs. We also conducted a series of experiments to identify the optimal number of SVs for our algorithm, selected from $\{3, 4, \ldots, 15\}$. This yields significant accuracy, as given in Table 1. The computational cost, using the best configuration of the tunable parameters, are given in terms of CPU time[3].

Time series domain description

We also conducted some experiments on a time series. It consists of the variation in chlorine concentration at a given node in a water network. Chlorine is a highly efficient disinfectant, injected in water supplies to kill residual bacteria. This time series exhibits large fluctuations due to the variations in water consumption and an inefficient control system. This data, taken from the public water supplies of the Cannes city in France, was sampled at the rate of a sample every 3 minutes. We considered 4 days of chlorine concentration measures. See Figure 1.

To capture the structure of the time series, a 3-length sliding window was used, with $\boldsymbol{x}_i = [x_{i-2} \;\; x_{i-1} \;\; x_i]$, where the Gaussian kernel was applied. Only the first day, that is 481 samples, was considered for training and estimating the optimal parameters using a 10-fold cross-validation configuration with $\sigma \in \{2^{-5}; 2^{-4}; \cdots; 2^4; 2^5\}$. To be comparable with the one-class SVM, we considered $|\mathcal{I}| = 55$ for both methods, obtained from $\nu = 0.004$ in one-class SVM, leading to 60% of common SVs. Table 2 provides a comparative study, with training error given by the cross-validation, and the test error estimated on the next 3 days. This shows the relevance of our approach, where we also illustrate the upper bound on the approximation error derived in Theorem 1.

[3]CPU time estimated on a Matlab running on a 2.53 GHz Intel Core 2 Duo processor and 2 GB RAM.

	training error	time (m:ss)	coherence μ_0	bound in (4)	test error
one-class SVM	8.90 %	1:16	—	—	63.7 %
this paper	0.20 %	0:02	0.80	0.37	1.9 %

Table 2. Results obtained for the time series problem.

5. CONCLUSION

In this paper, we investigated a new one-class classification method, by using the coherence criterion. We derived an upper bound on the error of approximating the center of the sphere by the resulting reduced model. We incorporated this criterion into a new kernel-based one-class algorithm by solving a least-squares optimization problem, and considered an incremental and decremental schemes. Experiments were conducted on real datasets to compare our approach to existing methods. Perspectives include the use of this approach to derive an online one-class algorithm for novelty detection.

6. REFERENCES

[1] D. Tax, "One-class classification," PhD thesis, Delft University of Technology, Delft, June 2001.

[2] B. Schölkopf, J. C. Platt, J. 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.

[3] Y. Zhang, N. Meratnia, and P. Havinga, "Adaptive and online one-class support vector machine-based outlier detection techniques for wireless sensor networks," in *Proc. of the IEEE 23rd International Conference on Advanced Information Networking and Applications Workshops/Symposia*, Bradford, United Kingdom, May 2009, pp. 990–995.

[4] F. Ratle, M. Kanevski, A.-L. Terrettaz-Zufferey, P. Esseiva, and O. Ribaux, "A comparison of one-class classifiers for novelty detection in forensic case data," in *Proc. 8th international conference on Intelligent data engineering and automated learning*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 67–76.

[5] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, March 2009.

[6] P. Honeine, C. Richard, and J. C. M. Bermudez, "On-line nonlinear sparse approximation of functions," in *Proc. IEEE International Symposium on Information Theory*, Nice, France, June 2007, pp. 956–960.

[7] D. Wang, D. S. Yeung, and E. C. C. Tsang, "Structured one class classification," *IEEE Trans. on systems, Man, and Cybernetics, Part B*, vol. 36, pp. 1283–1295, 2006.

[8] Y.-H. Liu, Y.-C. Liu, and Y.-J. Chen, "Fast support vector data descriptions for novelty detection," *IEEE Trans. on Neural Networks*, vol. 21, no. 8, pp. 1296–1313, aug. 2010.