Large-scale classification by an Approximate Least Squares One-Class Support Vector Machine ensemble

Vasileios Mygdalis, Alexandros Iosifidis, Anastasios Tefas and Ioannis Pitas
[†]Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece
Email: {tefas,pitas}@aiia.csd.auth.gr

***Abstract*—Large-scale multi-class classification problems involve an enormous amount of training data that make the application of classical non-linear classification algorithms difficult. In addition, such multi-class classification problems are usually formed by a considerable number of classes. This makes the application of the popular one-versus-rest binary classifiers fusion scheme adopted by most state-of-the-art approaches difficult. In this paper, in order to overcome the high computational cost of multi-class non-linear classification approaches, we adopt an ensemble of approximate non-linear one-class classifiers. To this end, we propose a new scalable solution for the Least Squares One-Class Support Vector Machine classifier by following an approximate kernel approach. We evaluated the proposed method in big data visual classification problems, where it is shown that it is able to achieve satisfactory performance, while significantly reducing the overall computational and memory costs.**

I. Introduction

In large-scale classification problems, training for multiple classes by employing state-of-the art non-linear classification methods, like the kernel Support Vector Machines variants [1], [2], [3], is computationally prohibitive due to the enormous number of training samples. For that case, a simple solution would be to employ one-class classification methods and model each class separately, this way reducing the training set cardinality of each smaller (one-class) classification problem significantly. Moreover, a recently concluded study has indicated that employing an ensemble of one-class classifiers can achieve very good performance in large-scale multi-class biomedical data classification problems [4]. Novelty detection methods are commonly used when only the class of interest needs to be modelled and discriminated from the rest of the world (every other possibility). Common use case scenarios for novelty detection methods include the case where only one class is well sampled and at the same time more important than every other possibility, such as in medical diagnostic problems, fault detection, video surveillance, mobile fraud detection [5], as well as on video summarization [6].

State-of-the-art one-class classification methods, including the One-class Support Vector Machines [7] (OC-SVM), the Support Vector Data Description [8] (SVDD), kernel PCA-based methods [9], [10] and the recently proposed Least Squares One-Class Support Vector Machine [11], that can derive non-linear solutions by exploiting the well-known kernel trick [12], [13] achieve significantly better performance over their linear alternatives. Such non-linear methods exploit the so-called kernel matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$, where N is the number of training data, in order to define linear solutions in the so-called kernel space $\mathcal{F}$ (of arbitrary dimensionality), which correspond to non-linear solutions in the original (input) space. The derived solutions usually involve the eigen-decomposition or the inversion of $\mathbf{K}$. In classification problems involving Big Data, where N very large, the application of such approaches is prohibitive, since their computational complexity scales as $O(N^3)$ and their memory complexity scales as $O(N^2)$.

In order to overcome these restrictions of kernel methods, approaches exploiting low-rank matrix approximation methods have been proposed [14], [15], [16]. A popular approach in this category of methods exploits a low-rank approximatation $\tilde{\mathbf{K}} \in \mathbb{R}^{N \times N}$ of the kernel matrix $\mathbf{K}$ which can be obtained by random (column) sampling of $\mathbf{K}$ and is given by $\mathbf{K} \approx \mathbf{C}\mathbf{W}^{\dagger}\mathbf{C}^{T}$, where $\mathbf{C} \in \mathbb{R}^{N \times m}$ contains the m sampled columns and $\mathbf{W} \in \mathbb{R}^{m \times m}$ is the kernel matrix of the training data corresponding to the m sampled columns. By following such an approach, a reduced number of data similarities need to be calculated and stored, leading to lower computational complexity and lower memory requirements. However, employing low-rank matrix approximations for any kernel method, including the Kernel Ridge Regression [1] (KRR), as well as equivalent methods like Regularization Network [2] (RR), Least-Squares Support Vector Machines [3] and Extreme Learning Machines [17], [18], may not be the most optimal approximation solution in every case.

In order to find an approximate solution for the Least Squares One-Class Support Vector Machine (LSOCSVM) classifier, we should follow an approach similar to other approximate methods [19], [20]. For instance, an approximate solution of the kernel K-means optimization problem is obtained by approximating the cluster centers using similarities between randomly sampled points and all points of the data [19]. In Support Vector Machines, since the decision hyperplane is defined by the support vectors, which are expected to be lesser than the training data, a method that approximates the extreme points (that are more likely to be the support vectors) has been proposed in [20]. This work has shown that method-specific approximate solutions are also an option, besides kernel matrix approximation ones.

In this paper, in order to perform large-scale visual data classification, we propose a multi-class classification scheme that employs an ensemble of One-class classifiers. We employ the one-class classifiers in order to model each class independently. In order to avoid heavy computations in these

one-class classification problems (since in Big Data problems the number of samples forming the various classes can be again high), we propose a new approximate solution for the LSOCSVM classifier, called Approximate Least Squares One-Class Support Vector Machine (ALSOCSVM) classifier. We evaluate the proposed approach in multi-class face recognition and activity recognition classification problems.

The remainder of the paper is structured as follows. In Section II we provide an overview of the Least Squares One Class Support Vector Machine (LSOCSVM) classifier. In Section III, we describe in detail the proposed Approximate Least Squares One Class Support Vector Machine (ALSOCSVM) classifier. Experiments conducted in order to test our approach are described in Section IV. Finally, conclusions are drawn in Section V.

II. Least Squares One Class Support Vector Machine

Let us denote by $\mathcal{X} = \{\mathbf{x}_i, \ldots, \mathbf{x}_N\}$ a set of training data $\mathbf{x}_i \in \mathbb{R}^D, i = 1, \ldots, N$ that form a distinct class in a multi-class classification problem. In order to derive non-linear decision functions, we can exploit any non-linear function $\phi(\cdot)$ such that $\mathbf{x}_i \in \mathbb{R}^D \rightarrow \phi(\mathbf{x}_i) \in \mathcal{F}$ in order to map the data from the input space $\mathbb{R}^D$ to the feature space $\mathcal{F}$ (which can be of arbitrary dimensions in the RBF case). Let us denote by $\mathbf{\Phi} = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)]$ a matrix (of arbitrary dimensions) that contains the training data representations in $\mathcal{F}$. Let us also define by $\mathbf{K} = \mathbf{\Phi}^T\mathbf{\Phi}$ the kernel matrix having elements $[\mathbf{K}]_{ij} = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$.

The Least Squares One-Class Support Vector Machine classifier [11] solves the following optimization problem[1] for the calculation of the optimal separating hyperplane $\mathbf{w} \in \mathcal{F}$:

$$\begin{aligned} &Minimize && \tfrac{1}{2}\|\mathbf{w}\|_2^2 + \tfrac{c}{2}\textstyle\sum_{i=1}^N \xi_i^2, && (1)\\ &Subject\ to && \mathbf{w}^T\phi(\mathbf{x}_i) = 1 - \xi_i, \quad i = 1, ..., N, && (2)\end{aligned}$$

leading to the solution:

$$\mathbf{w} = \mathbf{\Phi}\left(\mathbf{\Phi}^T\mathbf{\Phi} + \frac{1}{c}\mathbf{I}\right)^{-1}\mathbf{1} = \mathbf{\Phi}\left(\mathbf{K} + \frac{1}{c}\mathbf{I}\right)^{-1}\mathbf{1}, \quad (3)$$

where $\mathbf{1} \in \mathbb{R}^N$ is a vector of ones and c a parameter that denotes the importance of training error. Since the feature space $\mathcal{F}$ is usually unknown, $\mathbf{w}$ in (3) cannot be directly calculated. By exploiting the Representer Theorem [21], [22], [23], $\mathbf{w}$ is usually restricted to belong to the span of $\mathbf{\Phi}$, i.e. $\mathbf{w} = \mathbf{\Phi}\mathbf{a}$, where $\mathbf{a} \in \mathbb{R}^N$. From (3), we obtain:

$$\mathbf{a} = \left(\mathbf{K} + \frac{1}{c}\mathbf{I}\right)^{-1}\mathbf{1}. \quad (4)$$

After the calculation of $\mathbf{a}$, the decision for a new (unknown) sample $\mathbf{x}_t$ can be given by:

$$o_t = \mathbf{w}^T\phi(\mathbf{x}_t) = \mathbf{a}^T\mathbf{\Phi}^T\phi(\mathbf{x}_t) = \mathbf{a}^T\mathbf{k}_t, \quad (5)$$

where $\mathbf{k}_t \in \mathbb{R}^N$ is a vector having its elements equal to $[\mathbf{k}_t]_i = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_t),\ i = 1, \ldots, N$.

[1]This is a slightly modified version of the optimization problem solved by LSOCSVM classifier that leads to a non-normalized solution. The original LSOCSVM solution is similar to that in (4) by exploiting an additional normalization term

III. Approximate Least Squares One-Class Support Vector Machine

The Approximate Least Squares One-Class SVM classifier, similar to the LSOCSVM one, maps the training data $\mathbf{x}_i,\ i = 1, \ldots, N$ to the kernel space $\mathcal{F}$ by using the non-linear function $\phi(\cdot)$ and solves the optimization problem in (1) under the constraints in (2). However, we restrict the solution to be in the range of a subset of the training data, forming the matrix $\tilde{\mathbf{\Phi}} \in \mathbb{R}^{|\mathcal{F}|\times n}$, where $n < N$. That is, the ALSOCSVM classifier seeks for a solution of the form:

$$\tilde{\mathbf{w}} = \tilde{\mathbf{\Phi}}\mathbf{a}. \quad (6)$$

More formally, the ALSOCSVM classifier solves the following optimization problem:

$$\begin{aligned} &Minimize && \frac{1}{2}\tilde{\mathbf{w}}^T\tilde{\mathbf{w}} + \frac{c}{2}\sum_{i=1}^N \xi_i^2, && (7)\\ &Subject\ to && \tilde{\mathbf{w}}^T\phi(\mathbf{x}_i) = 1 - \xi_i, \quad i = 1, ..., N. && (8)\end{aligned}$$

For notational convenience, let us denote the auxiliary matrices $\mathbf{S} \in \mathbb{R} = \tilde{\mathbf{\Phi}}^T\tilde{\mathbf{\Phi}}^{n\times n}$ and $\tilde{\mathbf{K}} = \tilde{\mathbf{\Phi}}^T\mathbf{\Phi} \in \mathbb{R}^{n\times N}$. The equivalent to (7) as a dual optimization problem is given by:

$$\mathcal{L}(\mathbf{a}, \xi_i, \lambda_i) = \frac{1}{2}\mathbf{a}^T\mathbf{S}\mathbf{a} + \frac{c}{2}\sum_{i=1}^N \xi_i^2 - \sum_{i=1}^N \lambda_i(\mathbf{a}^T\mathbf{k}_i - 1 + \xi),$$

where $\lambda_i,\ i = 1, \ldots, N$ are the Lagrangian multipliers corresponding to the constraints in (8).

By determining the saddle points of the Lagrangian (9), we obtain:

$$\begin{aligned} \frac{\vartheta\mathcal{L}(\mathbf{a}, \xi_i, \lambda_i)}{\vartheta\mathbf{a}} &= 0 \Rightarrow \mathbf{S}\mathbf{a} = \tilde{\mathbf{K}}\boldsymbol{\lambda} && (9)\\ \frac{\vartheta\mathcal{L}(\mathbf{a}, \xi_i, \lambda_i)}{\vartheta\xi_i} &= 0 \Rightarrow \boldsymbol{\xi} = \frac{1}{c}\boldsymbol{\lambda} && (10)\\ \frac{\vartheta\mathcal{L}(\mathbf{a}, \xi_i, \lambda_i)}{\vartheta\lambda_i} &= 0 \Rightarrow \tilde{\mathbf{K}}^T\mathbf{a} = \mathbf{1} - \boldsymbol{\xi}. && (11)\end{aligned}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^N$ is a vector containing the slack variables λ_i.

By substituting (10) in (11), we obtain:

$$\boldsymbol{\lambda} = c(\mathbf{1} - \tilde{\mathbf{K}}^T\mathbf{a}) \quad (12)$$

Then, by substituting (12) in (9) we obtain:

$$\mathbf{a} = \left(\frac{1}{c}\mathbf{S} + \tilde{\mathbf{K}}\tilde{\mathbf{K}}^T\right)^{-1}\tilde{\mathbf{K}}\mathbf{1}. \quad (13)$$

By observing (13), we can see that the solution of the ALSOCSVM classifier requires the calculation and inversion of a $n \times n$ matrix, thus, highly reducing both the computational and memory costs (when compared to the LSOCSVM case). Here we can also observe that for a value of $n = N$, $\mathbf{S} = \tilde{\mathbf{K}} = \mathbf{K}$, and the solution of the (13) is exactly equal to (4):

$$\begin{aligned} \left(\frac{1}{c}\mathbf{K} + \mathbf{K}\mathbf{K}\right)\mathbf{a} &= \mathbf{K}\mathbf{1},\\ \mathbf{a} &= \left(\frac{1}{c}\mathbf{I} + \mathbf{K}\right)^{-1}\mathbf{1}. && (14)\end{aligned}$$

That is, the solution of the LSOCSVM can be approximated by using higher values of n.

After the calculation of $\mathbf{a}$, the decision for a new (unknown) sample $\mathbf{x}_t$ is given by:

$$o_t = \mathbf{w}^T \phi(\mathbf{x}_t) = \mathbf{a}^T \tilde{\mathbf{\Phi}}^T \phi(\mathbf{x}_t) = \mathbf{a}^T \tilde{\mathbf{k}}_t, \quad (15)$$

where $\tilde{\mathbf{k}}_t \in \mathbb{R}^n$ is the kernel vector obtained by using the training data forming $\tilde{\mathbf{\Phi}}$.

Finally, $\mathbf{x}_t$ is classified to the class under consideration if $(o_t - 1)^2 \leq \epsilon$, or is characterized as an outlier, if $(o_l - 1)^2 > \epsilon$, where $\epsilon \geq 0$ is a threshold determined by using the decision values obtained for the training data.

In terms of computational complexity, we can compare the solutions of LSOCSVM (4) with the solution of the proposed ALSOCSVM (15). We observe that the solution of LSOCSVM requires:

- the kernel matrix calculation, which has a computational complexity of $O(DN^2)$,
- the inversion of an $N \times N$ matrix, which has a computational complexity of the order of $O(N^3)$ and
- the calculation of $\mathbf{a}$, which has a computational complexity of $O(N^2)$.

The overall computational complexity of the LSOCSVM classifier in the training phase is equal to $O(N^3 + (D+1)N^2)$, while its computational complexity at the test phase is equal to $O(DN)$.

The solution of the ALSOCSVM requires:

- the calculation of $\tilde{\mathbf{K}}$ and $\mathbf{S}$, which has a computational complexity of $O(nND + n^2)$,
- the inversion of an $n \times n$ matrix, which has a computational complexity of the order of $O(n^3)$ and
- the calculation of $\mathbf{a}$, which has a computational complexity of $O(n^2 N)$.

The overall computational complexity of the ALSOCSVM classifier in the training phase is equal to $O(n^2(N+1) + nND)$, while its computational complexity at the test phase is equal to $O(nD)$.

From the above, we can observe that the computational complexity of the ALSOCSVM classifier is very low, when compared to that of the LSOCSVM classifier.

IV. Experimental Results

In this section, we describe experiments carried out in order to evaluate the performance of the proposed approach. We have evaluated the proposed ensemble of ALSOCSVM classifiers in three publicly available datasets, two for action recognition and one for face recognition. For comparison reasons, we have also trained an ensemble of one-class Support Vector Machines [7] (OC-SVM), Nystrom-based Approximate OC-SVMs [7], [14] (AOC-SVM) and Nystrom-based Approximate One-class Kernel Ridge Regressor [24], [14] (NY-OCKRR) classifiers. For evaluating the performance of each approach, in all experiments we have employed the g-mean metric [25] that can be used in classification problems as the ones taken into account in this paper. For all methods, we report the mean performance obtained over all classes. For the approximate methods, for a specific p value we randomly sample $n = pN$ training vectors to form the subset of data used for the calculation of the corresponding matrices, e.g. $\mathbf{S}$ and $\tilde{\mathbf{K}}$ for ALSOCSVM, and train the classifier, which is tested on the test data. Since all approximate methods employ random subsets of the training data, we repeat this process 10 times and calculate the mean performance and the corresponding standard deviation. All experiments were conducted on an Intel i7-3.6Ghz CPU with 32GB of RAM, using a MATLAB implementation. Specifically for the OC-SVM and AOC-SVM classifiers, we have employed the LIBSVM library [26], which is written in C++. Experiments for each visual data classification problem are given separately in subsections IV-1 and IV-2.

1) Experiments in face recognition: For performing experiments in face recognition, we have employed the YouTube faces dataset [27]. Example images can be seen in Figure 1. Vectorial image representations $\mathbf{x}_i \in \Re^D$, were obtained

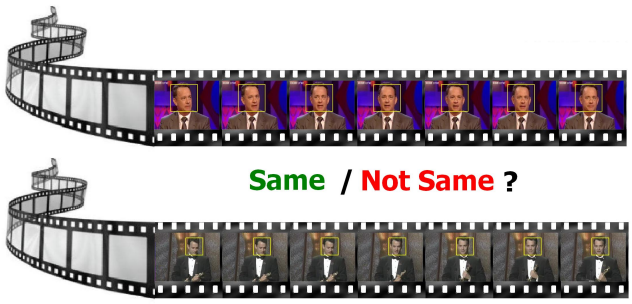

Fig. 1. Example faces from the YouTube faces dataset

by employing the Local Binary Pattern (LBP) descriptors, as in [27], leading to data dimensionallity $D = 1770$. In this experiment, we have kept only the classes that contained more than 1000 examples, resulting in a total 222183 images. We have performed a 10-fold cross-validation procedure, by splitting the dataset into 199973 train images and 22210 test images for each fold. In order to create balanced classification sets, we performed testing for each class separately using all test examples from the modelled class and a random selection from the 22210 test samples of twice as more elements.

In Table I, we provide the g-mean rates obtained by applying the competing algorithms in Youtube Faces dataset. For all approximate methods we provide the average g-mean rate and the corresponding standard deviation over ten experiments. As can be seen, satisfactory performance performance is achieved for all values of p. Moreover, state-of-the-art performance comparable to OC-SVM and LS-OC-SVM, can be obtained from a value of $p > 0.02$, for any approximation method. The performance of all approximate methods is similar for most of the values of p tested.

The time required to train each one-class classifier on the one-class problem corresponding to the class formed by the higher number of samples ($N = 2500$) is shown in Table II. Since OC-SVM and AOC-SVM classifiers were implemented using LIBSVM [26], which is a C++ implementation, they are expected to operate faster that LSOCSVM. Although

TABLE I. PERFORMANCE AND STDS ON YOUTUBE FACES DATASET

p/method	LSOCSVM	OC-SVM	AOC-SVM	NY-OCKRR	**ALSOCSVM**
0.01	-	-	**93.60 ± 11.31**	92.27 ± 7.07	92.79 ± 6.93
0.02	-	-	94.86 ± 8.60	95.22 ± 3.89	**95.29 ± 3.80**
0.05	-	-	96.19 ± 3.86	96.01 ± 1.92	**96.25** ± 1.90
0.1	-	-	**96.71** ± 1.49	96.15 ± 1.39	96.34 ± 1.21
0.2	-	-	**96.85** ± 1.03	96.01 ± 1.14	96.26 ± 0.88
0.4	-	-	**96.73** ± 0.72	95.90 ± 0.73	96.13 ± 0.53
1.0	96.11	**96.32**	**96.32**	96.11	96.11

the proposed ALSOCSVM classifier was implemented using strictly MATLAB code, it operates faster than the competition for values of $p < 0.4$. Moreover, the proposed ALSOCSVM classifier scales very well against the NY-OCKRR, which is main competition, and manages to have similar training times with LSOCSVM, for values of p close to 1.

TABLE II. TRAINING TIMES (SECONDS) REQUIRED FOR A SINGLE CLASS

p/method	LSOCSVM	OC-SVM*	AOC-SVM*	NY-OCKRR	**ALSOCSVM**
0.01	-	-	0.17	0.06	**0.02**
0.02	-	-	0.18	0.07	**0.02**
0.05	-	-	0.19	0.09	**0.04**
0.1	-	-	0.23	0.17	**0.07**
0.2	-	-	0.36	0.39	**0.17**
0.4	-	-	**0.61**	1.3	0.68
0.6	-	-	**0.91**	3.75	1.8
0.8	-	-	**1.16**	8.19	4.3
1.0	7.1	**0.4**	1.5	15.7	7.5

2) Experiments in Action Recognition: For action recognition, we have employed the outdoor recording section of the IMPART Multi-modal/Multi-view Dataset of [28] and the i3DPost Multi-view Human Action Dataset [29]. IMPART outdoor recording section consists of synchronized recordings of 14 cameras, depicting 3 actors performing 9 activities. We obtained segmented videos depicting each activity by applying the activity video segmentation method [30]. Example frames from the IMPART dataset can be seen in Figure 2. The

Fig. 2. Sample frames from the IMPART dataset

i3DPost dataset contains 832 high-resolution (1080×1920 pixel) videos depicting 8 persons performing 13 activities. Example frames from the I3DPost dataset can be seen in Figure 3. For both datasets, we have employed the Dense Trajectory-

Fig. 3. Sample frames from the I3DPost dataset

based video description [31], which has shown to provide state-of-the-art performance in human action recognition. We obtained vectorial video representations be employing the Bag-of-Features representation [32], [33], leading to vectors of 100 dimensions for each video. We performed a 3-fold cross-validation procedure, where we used 2/3 of the videos for training and the remaining 1/3 of them for testing.

Experimental results in terms of classification performance are shown in Tables III and IV. As can be seen, the LSOCSVM classifier provides the best classification performance in both datasets. Comparable performance to LSOCSVM, can be obtained by employing the NY-OCKRR and ALSOCSVM classifiers for a value of $p > 0.3$. In almost every case, the NY-OCKRR and ALSOCSVM classifier provide similar classification performance.

TABLE III. PERFORMANCE IN I3DPOST DATASET

p/method	LSOCSVM	OC-SVM	AOC-SVM	NY-OCKRR	**ALSOCSVM**
0.01	-	-	**67.88**	65.36	65.81
0.02	-	-	**67.37**	65.95	65.60
0.05	-	-	**70.58**	68.97	69.25
0.1	-	-	**76.07**	75.06	75.19
0.2	-	-	78.33	78.61	**79.03**
0.3	-	-	79.52	**80.73**	80.42
0.5	-	-	77.88	82.75	**82.91**
0.7	-	-	76.09	**84.14**	84.04
0.8	-	-	75.19	**84.62**	84.43
1	**85.25**	73.78	73.78	**85.25**	**85.25**

TABLE IV. PERFORMANCE IN IMPART DATASET

p/method	LSOCLSVM	OC-SVM	AOC-SVM	NY-OCKRR	**ALSOCSVM**
0.01	-	-	**58.19**	56.73	56.56
0.02	-	-	**58.76**	56.63	56.03
0.05	-	-	**62.02**	60.13	60.15
0.1	-	-	**65.07**	62.11	62.35
0.2	-	-	**65.04**	64.87	64.70
0.3	-	-	64.30	65.91	**66.01**
0.5	-	-	62.90	**67.58**	67.38
0.7	-	-	61.39	**68.30**	68.18
0.8	-	-	61.11	**68.55**	68.49
1	**68.98**	60.47	60.47	**68.98**	**68.98**

n Tables V and VI we present the training times for the one-class classification ensemble. The proposed ALSOCSVM classifier provides the best performance in terms of training times, for values of $p < 0.3$. We also can see that it scales better than the competing NY-OCKRR classifier, for all values of p.

TABLE V. TRAINING TIMES (MS) IN I3DPOST DATASET

p/method	LSOCSVM	OC-SVM*	AOC-SVM*	NY-OCKRR	**ALSOCSVM**
0.01	-	-	**1.92**	2.46	2.06
0.02	-	-	2.54	2.74	**2.27**
0.05	-	-	3.12	3.11	**2.86**
0.1	-	-	3.35	3.50	**3.01**
0.2	-	-	3.99	4.56	**3.72**
0.3	-	-	4.29	5.35	**4.23**
0.5	-	-	**5.39**	10.26	6.72
0.7	-	-	**6.38**	13.84	8.90
0.8	-	-	**6.24**	14.78	9.61
1	11.88	**4.43**	6.82	18.46	11.88

V. CONCLUSION

In this paper, we have described a new approximate method for large scale visual data classification, that employs an ensemble of one-class classifiers. The approximation scheme restricts the solution to be a linear combination of a subset of the training data in the feature space. The proposed approach has shown decent generalization performance in visual data

TABLE VI. TRAINING TIMES (MS) IN IMPART DATASET

p/method	LSOCLSVM	OC-SVM*	AOC-SVM*	NY-OCKRR	**ALSOCSVM**
0.01	-	-	7.70	7.72	**7.30**
0.02	-	-	8.36	8.14	**7.67**
0.05	-	-	9.40	9.29	**8.70**
0.1	-	-	10.52	10.69	**9.74**
0.2	-	-	12.11	13.81	**12.00**
0.3	-	-	**14.93**	18.66	15.72
0.5	-	-	**18.93**	27.4	22.24
0.7	-	-	**24.11**	40.45	30.39
0.8	-	-	**26.64**	47.33	35.39
1	45.03	**19.75**	31.74	62.69	45.03

classification problems, with reduced memory and computational costs, when compared to other approximation choices. We expect that this method could also provide satisfactory performance in other classification problems. This will be investigated in our future research.

ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 316564 (IMPART).

REFERENCES

[1] C. Saunders, A. Gammerman, and V. Vovk, "Ridge regression learning algorithm in dual variables," in *(ICML-1998) Proceedings of the 15th International Conference on Machine Learning*. Morgan Kaufmann, 1998, pp. 515–521.

[2] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," *Advances in computational mathematics*, vol. 13, no. 1, pp. 1–50, 2000.

[3] I. Steinwart, "Consistency of support vector machines and other regularized kernel classifiers," *Information Theory, IEEE Transactions on*, vol. 51, no. 1, pp. 128–142, 2005.

[4] B. Krawczyk and M. Woniak, "Handling label noise in microarray classification with one-class classifier ensemble," in *ICT Innovations 2014*, ser. Advances in Intelligent Systems and Computing, A. M. Bogdanova and D. Gjorgjevikj, Eds. Springer International Publishing, 2015, vol. 311, pp. 351–359.

[5] M. Pimentel, D. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.

[6] V. Mygdalis, A. Iosifidis, A. Tefas, and I. Pitas, "Video summarization based on subclass support vector data description," *IEEE Symposium Series on Computational Intelligence*, 2014.

[7] B. Scholkopf, 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.

[8] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[9] H. Hoffmann, "Kernel pca for novelty detection," *Pattern Recognition*, vol. 40, no. 3, pp. 863–874, 2007.

[10] P. Bodesheim, A. Freytag, E. Rodner, M. Kemmler, and J. Denzler, "Kernel null space methods for novelty detection," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 3374–3381.

[11] Y. Choi, "Least squares one-class support vector machine," *Pattern Recognition Letters*, vol. 30, no. 13, pp. 1236–1240, 2014.

[12] V. Vapnik, *Statistical Learning Theory*. Wiley, New York, NY, 1998.

[13] R. Duda, P. Hart, and D. Stork, *Pattern Classification, 2nd ed.* Wiley-Interscience, 2000.

[14] D. Achlioptas and F. McSherry, "Fast computation of low rank matrix approximations," in *Proceedings of the thirty-third annual ACM symposium on Theory of computing*. ACM, 2001, pp. 611–618.

[15] P. Drineas and M. W. Mahoney, "On the nyström method for approximating a gram matrix for improved kernel-based learning," *The Journal of Machine Learning Research*, vol. 6, pp. 2153–2175, 2005.

[16] K. Zhang, I. W. Tsang, and J. T. Kwok, "Improved nyström low-rank approximation and error analysis," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1232–1239.

[17] G. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme Learning Machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.

[18] A. Iosifidis, A. Tefas, and I. Pitas, "On the kernel extreme learning machine classifier," *Pattern Recognition Letters*, vol. 54, pp. 11–17, 2015.

[19] R. Chitta, R. Jin, T. C. Havens, and A. K. Jain, "Approximate kernel k-means: Solution to large scale kernel clustering," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 895–903.

[20] M. Nandan, P. P. Khargonekar, and S. S. Talathi, "Fast svm training using approximate extreme points," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 59–98, 2014.

[21] K. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neur. Netw.*, vol. 12, no. 2, pp. 181–201, 2001.

[22] B. Scholkpf and A. Smola, *Learning with Kernels*. MIT Press, 2001.

[23] A. Argyriou, C. Micchelli, and M. Pontil, "When is there a representer theorem? vector versus matrix regularizers," *J. Mach. Learn. Res.*, vol. 10, pp. 2507–2529, 2009.

[24] Q. Leng, H. Qi, J. Miao, W. Zhu, and G. Su, "One-class classification with extreme learning machine," *Mathematical Problems in Engineering, Article ID 412957*, pp. 1–12, 2014.

[25] M. Kubat, R. C. Holte, and S. Matwin, "Machine learning for the detection of oil spills in satellite radar images," *Machine learning*, vol. 30, no. 2-3, pp. 195–215, 1998.

[26] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[27] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 529–534.

[28] H. Kim and A. Hilton, "Influence of colour and feature geometry on multi-modal 3d point clouds data registration," in *3D Vision (3DV), 2014 2nd International Conference on*, vol. 1. IEEE, 2014, pp. 202–209.

[29] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, and I. Pitas, "The i3dpost multi-view and 3d human action/interaction database," in *Visual Media Production, 2009. CVMP'09. Conference for*. IEEE, 2009, pp. 159–168.

[30] N. Kourous, A. Iosifidis, A. Tefas, N. Nikolaidis, and I. Pitas, "Video characterization based on activity clustering," in *Electrical and Computer Engineering (ICECE), 2014 International Conference on*. IEEE, 2014, pp. 266–269.

[31] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 60–79, 2013.

[32] Y. Huang, Z. Wu, L. Wang, and T. Tan, "Feature coding in image classification: A comprehensive study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 493–506.

[33] A. Iosifidis, A. Tefas, and I. Pitas, "Discriminant bag of words based representation for human action recognition," *Pattern Recognition Letters*, 2014.