Efficient Background Modeling through Incremental Support Vector Data Description

Alireza Tavakkoli and Mircea Nicolescu and George Bebis
Computer Vision Laboratory
University of Nevada, Reno
Email: {tavakkol,mircea,bebis}@cse.unr.edu

Monica Nicolescu
Robotics Laboratory
University of Nevada, Reno
Email: monica@cse.unr.edu

Abstract

Background modeling is an essential and important part of many high-level video processing applications. Recently, the Support Vector Data Description (SVDD) has been introduced for novelty detection when only one class of data is available, i.e. background pixels. This paper proposes a method to efficiently train an SVDD and compares the performance of this training algorithm with the traditional SVDD training techniques. We compare the performance of our method with traditional SVDD and other classification algorithms on various data sets including real video sequences.

1 Introduction

Background modeling is one of the most effective and widely used techniques to detect moving objects in videos with a quasi-stationary background. In these scenarios, despite the presence of a static camera, the background is not completely stationary due to inherent changes, such as water fountains, waving flags, etc. Statistical modeling approaches estimate the probability density function of the background pixel values.

Parametric density estimation methods, such as Mixture of Gaussians techniques (MoG) are proposed in [4]. If the data is not drawn from a mixture of normal distributions the parametric density estimation techniques may not be useful. As an alternative, non-parametric density estimation approaches can be used to estimate the probability of a given sample belonging to the same distribution function as the data set [5]. However, the memory requirements of the non-parametric approach and its computational costs are high since they require the evaluation of a kernel function for all data samples.

Support Vector Data Description (SVDD) is a technique which uses support vectors in order to model a data set [6]. The SVDD represents one class of known data samples in such a way that for a given test sample it can be recognized as known, or rejected as novel.

In this paper we present a novel incremental learning scheme to train SVDDs. Optimization converges by optimizing only on two data points with a specific condition [1] which requires at least one of the data points does not satisfy the KKT conditions – The conditions by which the classification requirements are satisfied – [3]. Our experimental results show that the our SVDD training achieves higher speed and require less memory than the online [7] and the canonical training [6].

The rest of the paper is organized as follows. Section 2 discusses the methodology used in this paper for the training of SVDDs. In Section 3 a comprehensive quantitative and qualitative set of experiments is carried out to compare the proposed incremental SVDD with the online and canonical training algorithms. Section 4 concludes the paper and gives future directions of study.

2 Methodology

In order to discuss the proposed algorithm we first introduce the SVDD method and its application. Then, we present the our incremental training algorithm.

2.1 Support Vector Data Description

A normal data description gives a closed boundary around the data which can be represented by a hypersphere (i.e. $F(R, a)$) with center a and radius R, whose volume should be minimized. To allow the possibility of outliers in the training set, slack variables $\epsilon_i \geqslant 0$ are introduced. The error function to be minimized is:

$$F(R, a) = R^2 + C \sum_i \epsilon_i \|x_i - a\|^2 \leqslant R^2 + \epsilon_i \quad (1)$$

subject to:

$$\|x_i - a\|^2 \leqslant R^2 + \epsilon_i \quad \forall i. \quad (2)$$

In order to have a flexible data description kernel functions $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$ are used. After applying the kernel and using Lagrange optimization the SVDD function becomes:

$$L = \sum_i \alpha_i K(x_i, x_i) - \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \qquad (3)$$
$$\forall \alpha_i : 0 \leqslant \alpha_i \leqslant C$$

Only data points with non-zero α_i are needed in the description of the data set, therefore they are called *support vectors* of the description. After optimizing (3) the Lagrange multipliers should satisfy the normalization constraint $\sum_i \alpha_i = 1$.

Optimizing equation (3) is a Quadratic Programming (QP) problem. Generally the SVDD is used to describe large data sets. In such applications optimization via standard QP techniques becomes intractable. To address this issue several algorithms have been proposed which employ faster solutions to the above QP problem.

2.2 Incremental SVDD

Our incremental training algorithm is based on the theorem proposed by Osuna *et al.* in [3]. According to Osuna a large QP problem can be broken into series of smaller sub-problems. The optimization converges as long as at least one sample violates the KKT conditions.

In the incremental learning scheme, at each step we add one sample to the training working set consisting of only support vectors. Assume we have a working set which minimizes the current SVDD objective function for the current data set. The KKT conditions do not hold for samples which do not belong to the description. Thus, the SVDD converges only for the set which includes a sample outside the description boundary.

The smallest possible sub-problem consists of only two samples [1]. Since only the new sample violates the KKT conditions at every step, our algorithm chooses one sample from the working set along with the new sample and solves the optimization on them.

Solving the QP problem for two Lagrange multipliers can be done analytically. Because there are only two multipliers at each step, the minimization constraint can be displayed in 2-D. The two Lagrange multipliers should satisfy the inequality in (3) and the linear equality in the normalization constraint.

We first compute the constraints on each of the two multipliers. The two Lagrange multipliers should lie on a diagonal line in 2-D (equality constraint) within a rectangular box (inequality constraint). Without loss of generality we consider that the algorithm starts with finding the upper and lower bounds on α_2 which are $H = min(C, \alpha_1^{old} + \alpha_2^{old})$ and $L = max(0, \alpha_1^{old} + \alpha_2^{old})$, respectively. The new value for α_2^{new} is computed by finding the maximum along the direction given by the linear equality constraint:

$$\alpha_2^{new} = \alpha_2^{old} + \frac{E_1 - E_2}{K(x_2, x_2) + K(x_1, x_1) - 2K(x_2, x_1)} \qquad (4)$$

where E_i is the error in evaluation of each multiplier. The denominator in (4) is a step size (second derivative of objective function along the linear equality constraint). If the new value for α_2^{new} exceeds the bounds it will be clipped ($\hat{\alpha}_2^{new}$). Finally, the new value for α_1 is computed using the linear equality constraint:

$$\alpha_1^{new} = \alpha_1^{old} + \alpha_2^{old} - \alpha_2^{new} \qquad (5)$$

2.3 The Background Modeling Algorithm

The algorithm for background modeling using the incremental SVDD is composed of two modules. In the background modeling module the incremental learning scheme presented in section 2.2 uses the pixel values in each frame to train their corresponding SVDDs. In the background subtraction module the pixel value in the current frame is used in the trained SVDD to label it as a known (background) or novel (foreground) pixel. The parameter C in the SVDD accounts for the system tolerance. In all of our experiments we set $C = 0.1$ and $\sigma = 5$. The optimal value for these parameters can be estimated by cross-validation.

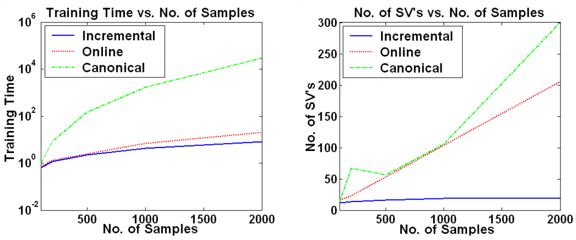
3 Experimental Results and Comparison

The experiments presented here are conducted in two main categories. The first set compares the performance of the proposed method in training the SVDDs with traditional methods on synthetic data sets. In the second set of experiments we show the performance of the proposed technique in a background modeling application.

3.1 Comparison on Synthetic Data

To show the performance and efficiency of the proposed method we compare our technique with the online [7] and batch SVDD [6] training algorithms. The experiments use Matlab 6.5 on a P4 Core Duo processor with 1GB RAM.

Speed and Number of Support Vectors. Figure 1(a) shows the training speed of our incremental SVDD, online and batch methods against the number of training samples. As seen, the proposed SVDD training runs faster than both batch and online algorithms. The online SVDD runs close to linear time but for larger data sets its training time is more than the proposed method.



(a) (b)

Figure 1. Comparison of the training algorithms on *banana* data set: (a) Training Speed. (b) No. of SV's.

Table 1. Comparison of the proposed training algorithm with, online and batch methods on three data sets of size 1000.

Data Set	Method	Error	F_1	# SV's	Time
Banana	Proposed	0.01	0.99	19	4.2
	Online	0.08	0.96	104	6.9
	Canonical	0.09	0.96	106	1697
Ellipse	Proposed	0.01	0.99	6	3.72
	Online	0.10	0.95	105	4.1
	Canonical	0.11	0.99	108	2314
Egg	Proposed	0.07	0.97	8	3.85
	Online	0.1	0.95	101	3.7
	Canonical	0.13	0.93	87	1581

A comparison of the number of retained support vectors for our technique and batch and online SVDD learning methods is presented in Figure 1(b). Our method keeps almost a constant number of support vectors by mapping to the same higher dimensional feature space for any given number of samples in the data set.

Error Evaluation. Table 1 compares the classification error, F_1 measure, number of the support vectors, and learning time for the three learning methods. The experiments are performed on three data sets ('*banana*', '*normal*', '*egg*') with 1000 training samples and 1000 test samples. The F_1 measure combines both the recall and the precision rates of a classifier:

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{6}$$

Classification Comparison. Table 2 compares the classification error, F_1 measure and asymptotic training and test time for various classifiers. As seen, the proposed training of the SVDDs reaches very good classification rates compared to other methods.

Classification Performance. In Figure 2(a) the classification boundaries of the three SVDD training algorithms are shown. The dots and circles are the training and test samples drawn from the *banana* data set, respectively. The $\star$, $\times$, and $+$ symbols are the support vectors of the incremental, online and canonical SVDDs, respectively. Figure 2(b) shows the Receiver Operating Curves (ROC). The solid curve is the ROC of the incremental learning while dotted and dashed curves correspond to the online and canonical methods.

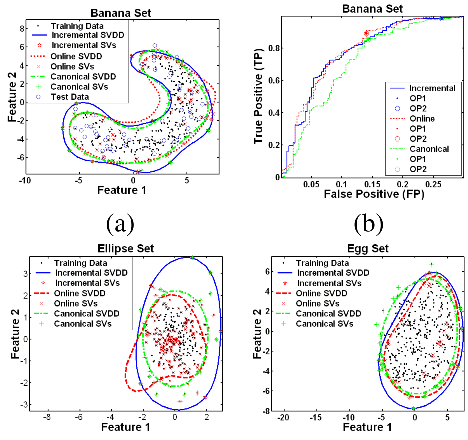
Figure 2(c) and (d) show a comparison of the classification boundaries between the three SVDD training algorithms on the 2-D *ellipse* data set and the 2-D *egg* data set, respectively. From Figure 2, the incremental SVDD results in more accurate classification boundaries than both online and canonical versions.

3.2 Application to Background Modeling

We applied the incremental SVDD (INCSVDD) to speed up the background modeling and compare its re-

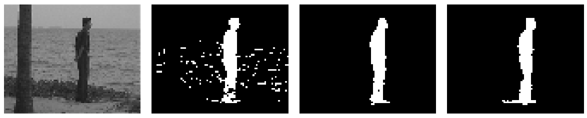
Table 2. Comparison of different classifiers on a *complex* data set.

Classifier	Error	F_1	Training	Test
Proposed	0.02	0.99	$O(1)$	$O(1)$
Batch SVD	0.10	0.95	$O(N)$	$O(N)$
Online SVD	0.10	0.94	$O(N)$	$O(N)$
KDE	0.12	0.94	$O(N)$	$O(N)$
MoG	0.14	0.92	$O(1)$	$O(1)$
K-means	0.15	0.92	$O(1)$	$O(1)$



(a) (b) (c) (d)

Figure 2. Comparison of methods: (a) *banana* data set. (b) Receiver Operating Curve (ROC) for *banana* data set. (c) *ellipse* data set. (d) *egg* data set.

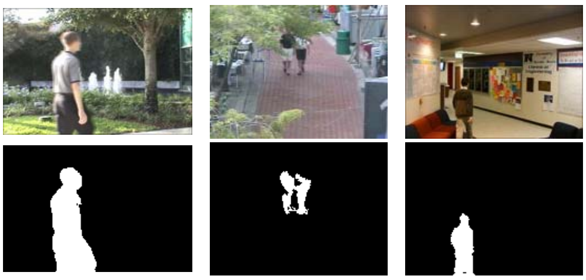


(a) (b) (c) (d)

Figure 3. Comparison of methods in presence of irregular motion in *water surface* video: (a) Original frame. (b) MoG results. (c) AKDE results. (d) INCSVDD results.

sults with traditional background modeling techniques.

Presence of Irregular Motion. We compare the results of foreground detection using our method with the AKDE [5] and MoG [4] on the *water surface* video. The results of MoG, the AKDE and our technique are shown in Figure 3 (b), (c), and (d), respectively.



(a) (b) (c)

Figure 4. Background modeling using the proposed method.

Difficult Scenarios. Figure 4 shows the results of foreground detection in videos using our method. The fountain, waving branches, and flickering lights in Figure 4(a), (b), and (c) pose significant challenges.

Table 3. Comparison of the proposed methods and traditional techniques.

Method	Classifier	Memory*	Time*
INSVDD	SVDD	$O(1)$	$O(1)$
AKDE[5]	Bayes	$O(N)$	$O(N)$
Spatio-temp[2]	Bayes	$O(N)$	$O(N)$
MoG[4]	Bayes	$O(1)$	$O(1)$
Wallflower[8]	K-means	$O(N)$	$O(N)$

$*$: Per-pixel
N: number of training frames

Comparison Summary. Table 3 shows a comparison of different traditional background modeling methods and our incremental SVDD (INCSVDD) technique. The only method which explicitly deal with the single-class classification is the proposed SVDD technique.

4 Conclusions

Many object recognition systems such as face detection can be represented as novelty detection applications since only one class of data samples (faces) may be available. Support Vector Data Descriptors (SVDD) can be employed in order to analytically model a single class of data. This paper proposes a method to efficiently train an SVDD based on incremental learning. The paper also presents a background modeling application in videos with quasi-stationary background using the proposed incremental SVDD. We showed the results of the proposed technique in a background modeling application and compared the system with traditional techniques, both quantitatively and qualitatively.

Acknowledgment

This work was supported by the NSF-EPSCoR Ring True III award EPS0447416 and by the Office of Naval Research award N00014-06-1-0611.

References

[1] J. Platt. Fast Training of Support Vector Machines using Sequential Minimal Optimization. *Advances in Kernel Methods - Support Vector Learning.*, MIT Press:185–208., 1998.

[2] L. Li, W. Huang, I. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing.*, 13(11):1459–1472, November 2004.

[3] E. Osuna, R. Freund, and F. Girosi. Improved Training Algorithm for Support Vector Machines. *In Proc. Neural Networks in Signal Processing*, 1997.

[4] C. Stauffer and W. Grimson. Learning Patterns of Activity using Real-Time Tracking. *IEEE Transactions on PAMI*, 22(8):747–757, August 2000.

[5] A. Tavakkoli, M. Nicolescu, and G. Bebis. Automatic Statistical Object Detection for Visual Surveillance. *In proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 144–148, March 2006.

[6] D. Tax and R. Duin. Support Vector Data Description. *Machine Learning*, 54(1):45–66., 2004.

[7] D. Tax and P. Laskov. Online SVM Learning: from Classification and Data Description and Back. *Neural Networks and Signal Processing*, (1):499–508., 2003.

[8] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: principles and practice of background maintenance. *In proceedings of ICCV*, 1:255–261, September 1999.