A Genetic Approach to Training Support Vector Data Descriptors for Background Modeling in Video Data

Alireza Tavakkoli, Amol Ambardekar, Mircea Nicolescu, and Sushil Louis

Department of Computer Science and Engineering
University of Nevada, Reno, USA
{tavakkol, ambardek, mircea, sushil}@cse.unr.edu

Abstract. Detecting regions of interest in video sequences is one of the most important tasks of most high level video processing applications. In this paper a novel approach based on Support Vector Data Description (SVDD) is presented which detects foreground regions in videos with quasi-stationary backgrounds. The SVDD is a technique used in analytically describing the data from a set of population samples. The training of Support Vector Machines (SVM's) in general, and SVDD in particular requires a Lagrange optimization which is computationally intensive. We propose to use a genetic approach to solve the Lagrange optimization problem more efficiently. The Genetic Algorithm (GA) starts with an initial guess and solves the optimization problem iteratively. We expect to get accurate results, moreover, with less cost than the traditional Sequential Minimal Optimization (SMO) technique.

1 Introduction

Stationary cameras are used in most visual surveillance systems. However, the background of the video may not be completely static, because of inherent changes in the background itself (fluctuations in monitors and lights, waving flags and/or trees, water surfaces, etc.). In these types of backgrounds a single background frame is not useful to detect moving regions.

A Mixture of Gaussians (MoG) modeling technique was proposed in [1] to address the multi-modality of the underlying background. Recently, a recursive filter formulation for MoG training is proposed by Lee in [2]. In [3], El Gammal *et al.* proposed a non-parametric Kernel Density Estimation (KDE) method for pixel-wise background modeling without making any assumption about its probability distribution. In order to adapt the model a sliding window is used in [4]. However, the model convergence is problematic in situations when the illumination suddenly changes. In methods that explicitly model the background, the foreground is detected by comparing each pixel model with a heuristically selected, global threshold [3], or locally trained thresholds [5].

In this paper a single-class classification approach is used to label pixels in video sequences into foreground and background classes using Support Vector

Data Description (SVDD) [6]. The SVDD is a technique which is used in describing the data analytically from a set of population samples [7]. This technique uses a generalized support vector learning scheme for novelty detection, when samples from outliers are not accessible by measurement [8]. Such samples in video surveillance applications are the foreground regions. These regions are not accessible in the training stage of the system.

To train the SVDD system Lagrange optimization, a quadratic programming (QP) optimization problem, must be solved. The most common technique to solve the QP problem is Sequential Minimal Optimization (SMO), proposed by Platt in [9]. However, the complexity of the SMO algorithm increases with the difference between the number of samples and their dimensionality.

Lessmaan *et.al* [10] proposed a model selection for Support Vector Machines (SVM) using a genetic algorithm (GA) approach. However, this technique deals with the models used to expand the SVM and addresses the classification accuracy. Liu *et.al* proposed a weighted SVM with a GA based parameter selection in [11]. Their method finds reliable weights for those samples with better support and finds the SVM parameters using a GA. However, these methods do not explicitly employ GA's as a tool for training the SVM.

The main contribution of this paper is to propose an evolutionary computing approach to solve the optimization problem in training of the SVDD. Our method encodes the description of samples as the genetic structure of individuals for a GA. This chromosome is then used to generate an evolving population using the proposed algorithm. The surviving genes of the fittest individual represents the support vectors of the sample set.

The rest of the paper is organized as follows. In section 2 we present a review of the SVDD. Section 3 and section 4 describe the GA approach and our proposed method, respectively. In section 5 experimental results of the proposed algorithm are provided and discussed. Finally, section 6 concludes this paper and gives future directions for this research.

2 Support vector data description

Data domain description concerns the characteristics of a data set [7] whose boundary can be used to detect novel samples (outliers). A normal data description gives a closed boundary around the data which can be represented by a hyper-sphere (i.e. $F(R, a)$). The volume of this hyper-sphere with center a and radius R should be minimized while containing all the training samples x_i. As proposed in [7] the extension to more complex distributions is straightforward using kernels.

To allow the possibility of outliers in the training set, slack variables $\epsilon_i \geq 0$ are introduced. The error function to be minimized is defined as $F(R, a) = R^2 + C\sum_i \epsilon_i$, subject to: $\|x_i - a\|^2 \leq R^2 + \epsilon_i$ for all i. Using Lagrange optimization the above results in:

$$L = \sum_i \alpha_i (x_i \cdot x_i) - \sum_{i,j} \alpha_i \alpha_j (x_i \cdot x_j) \quad \forall \alpha_i : 0 \leq \alpha_i \leq C \tag{1}$$

When a sample falls in the hyper-sphere then its corresponding Lagrange multiplier is $\alpha_i \geq 0$, otherwise it is zero. It can be observed that 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. To test a new sample y, its distance to the center of the hyper-sphere is calculated and tested against R. Given the support vectors x_i, a new test sample z_t can be classified as known/novel data using:

$$\|z_t - a\|^2 = (z_t \cdot z_t) - 2\sum_i \alpha_i (z_t \cdot x_i) + \sum_{i,j} \alpha_i \alpha_j (x_i \cdot x_j) \tag{2}$$

where α_i are Lagrange multipliers and $||z_t - \mathbf{a}||$ is the distance of the new sample to the description center. If this distance is larger than R then the sample is classified as novel.

In order to have a flexible data description, as opposed to the simple hyper-sphere discussed above, a kernel function $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$ is introduced. This kernel maps the data into a higher dimensional space, where it is described by the simple hyper-sphere boundary. Instead of a simple dot product of the training samples $(x_i \cdot x_j)$, the dot product is performed using a kernel function.

Several kernels have been proposed in the literature [12]. Among these, the Gaussian kernel gives a closed data description, $K(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right)$.

Using the above theory the proposed method generates a SVDD for each pixel in the scene using its values in the past. These descriptions are then used to label each pixel in new frames as a known (background) or a novel (foreground) pixel. In the following section we present the motivations behind using a GA to solve the SVDD training problem.

3 The Genetic Algorithm approach

The theory behind using a SVDD is to find those sample points whose α Lagrange multipliers are not zero (support vectors). In order to find support vectors for a given distribution, an optimization problem should be solved which results in all non-zero Lagrange multipliers from equation (1).

We propose to use an evolutionary computing method to solve the optimization problem in training of the SVDD. An evolutionary algorithm is capable of adapting to near-optimal solutions efficiently.

For the sake of illustration we assume, without loss of generality, that the data is in 2-D generated using a normal distribution function. As discussed in the previous section, the extension to higher dimensions and more complex distributions is straightforward.

Lagrange multipliers are non-zero only for the support vectors. Support vectors describe the distribution of the data using equation (1). Solving the optimization problem with respect to $\mathbf{a}$ (the center of discription) results in $\mathbf{a} = \sum_i \alpha_i \mathbf{x}_i$. Note that the Lagrange multipliers should be normalized ($\sum_i \alpha_i = 1$).

In our approach the optimization problem is solved in a bottom-up fashion. We start with random initial values for the α multipliers. Given these multipliers, the corresponding data description is generated. The data description is

1. Generate populatoin: α_i's according to $\sum_i \alpha_i = 1$
2. For each individual in population:
   2.1. $\mathbf{a} = \sum \alpha_i \mathbf{x}_i$
   2.2. $R_i = ||\mathbf{x}_i - \mathbf{a}||$
   2.3. $R = \sum \alpha_i R_i$
   2.4. Fitness = Percentage of Data covered by: $(R, \mathbf{a})$
4. Perform selection operation
5. Perform recombination $(\sum_i \alpha_i = 1)$
3. Perform evaluation $(\sum_i \alpha_i = 1)$
6. If target rate not reached goto 2.
7. Produce the data description: $\{\alpha_i, \mathbf{x}_i\}_{\forall i:\alpha_i \neq 0}$

Fig. 1. The proposed algorithm.

refined to achieve the target description iteratively according to the proposed evolutionary technique.

4 The proposed method

The proposed evolutionary algorithm is presented in Fig. 1. The algorithm employs a genetic approach to generate the solution iteratively, where the target solution is the smallest circle encompassing the data points. The algorithm produces the best solution as soon as the target rate is reached, or after it is run for the maximum number of generations. In the following, we discuss the representation scheme, encoding of genomes, and selection strategies of the proposed GA, respectively.

4.1 Representation and encoding

The optimization results of an evolutionary method greatly depend on the representation scheme and encoding. A crucial issue in this context is choosing the individual chromosomes. A chromosome encodes the vital information about an individual and can be used to evaluate the fitness or the quality of that individual during evolution.

The goal in our algorithm is to find the best data description for an arbitrary distribution of samples. Each individual in the population is represented by the sample points and their corresponding α multipliers (Fig. 2). It can be observed that a data description is uniquely represented by support vectors x_i and their corresponding coefficients α_i. Thus, each individual's decoded chromosome represents a data description.

Given a number of training sample points, each evolving individual in the population can be represented by a vector containing the Lagrange multipliers α_i. The α values are real numbers between 0 and 1. Notice that the size of each chromosome is equal to the number of training samples. Our GA finds the combination of Lagrange multipliers that results in the target data description. Fig. 2 shows the representation and encoding schemes used in the proposed GA.

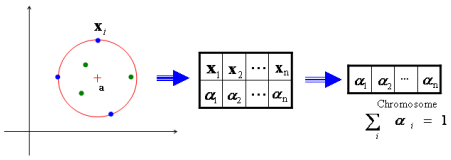


Fig. 2. Chromosome encoding and representation.

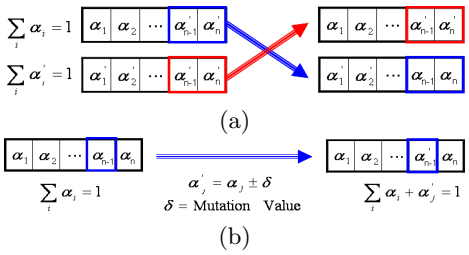


Fig. 3. Evolutionary operators: (a) Crossover operation. (b) Mutation operation.

4.2 Selection

In the proposed algorithm three different selection strategies are implemented. These strategies are roulette wheel selection, rank-based selection and $\mu\lambda$ selection. The effect of each strategy on the performance of the trained SVDD is studied.

Roulette wheel selection. This is a standard selection strategy used in canonical GA. This strategy explores the solution space more evenly. However, its convergence to the optimum in non-smooth search spaces is problematic. Since selection probability is proportional to the fitness the scaling problem is inevitable in this strategy [13].

Rank-based selection. This selection strategy addresses the problem of scaling. While being similar to the roulette wheel selection, individuals are selected with respect to their rank instead of their fitness value in this strategy.

$\mu\lambda$ **selection.** This selection strategy also addresses the scaling problem of canonical GA. $\mu\lambda$ is considered to be an elitist selection. In this strategy the best individual from the current population always survives to the next generation.

4.3 Crossover and mutation

It can be shown that crossover is the most important operator in GA's using the schema theorem. The crossover operator uses individuals in the population based on crossover probability and generates offsprings. The exploration power of a genetic algorithm relates to the type and probability of its crossover operator.

In our algorithm we used bi-parental crossover. From the selection stage we have N new individuals for a population size of N. Two parents are picked at random from the pool of individuals and are used in the crossover operation. This is done using the result of a biased toss of a coin (crossover probability).

This operator is similar to the crossover used in canonical GA except that it contains a string of real values. The procedure is summarized the best in Fig. 3(a). From the figure, the normalization condition is violated after the crossover. To maintain this condition both chromosomes are normalized.

Mutation explores the solution space to find better regions for optimization. A single point mutation is used in the proposed method. This operator picks a mutating gene randomly with the mutation probability using a biased coin toss. Fig. 3(b) depicts our proposed mutation operation. After the mutating gene is selected a small mutation bias value δ is added or subtracted. Since mutation violates the normalization of the multipliers the chromosome should be normalized.

5 Experimental results

We present two groups of experiments. The GA training experiments evaluate the performance of the proposed GA approach with regard to its selection strategies, population size and other significant parameters. The performance of the proposed method is compared with state-of-the-art techniques in the literature. The real data experiments compare the proposed system in detecting foreground regions in real videos to existing methods.

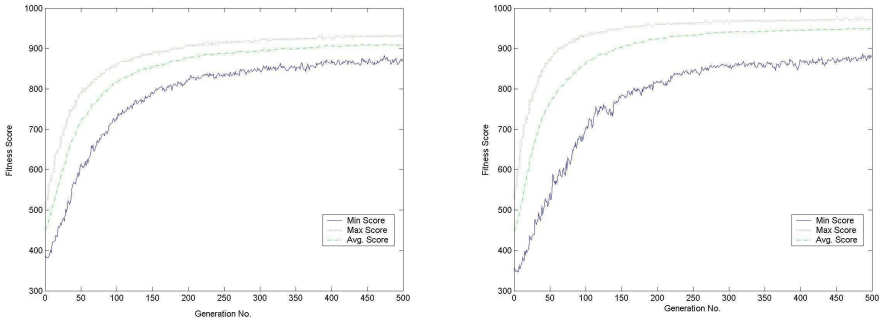
5.1 GA training experiments

A normal 2-D distribution of 100 data points is used for our experiments. As mentioned earlier the extension to higher dimensions and more complex distributions is straightforward, using kernels. In the following experiments we used a population size of 100, 0.5 for probability of crossover, 0.05 for probability of mutation, 30 runs, suppression parameter of 0.01, mutation bias of 0.01 and 500 generations, unless otherwise stated.

Effect of population size. In this experiment the effect of population size on the convergence and performance of the proposed approach is evaluated. The GA is run for 30 runs using the rank-based selection strategy.

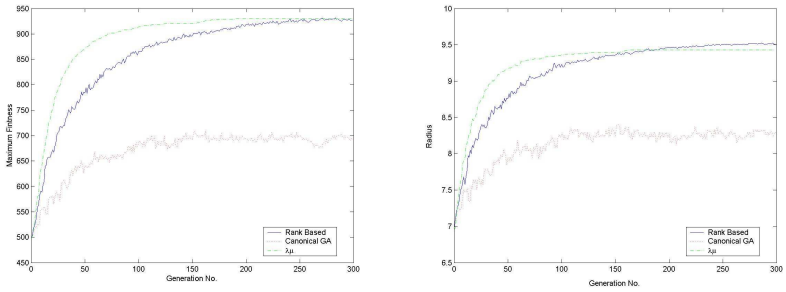
Fig. 4 shows the evolution of individuals and a comparison of the fitness values for population sizes of 20 and 100 individuals. In each graph the solid curve is the minimum, the dashed curve is the average and dotted curve represents the maximum fitness values over 30 runs. As it can be seen, the more individuals in the population, the faster the convergence of the GA. However, increasing the population size results in more calls to the evaluation function and a decrease in speed. Our experiments showed that a population size of 50 is best in terms of speed and accuracy.

Fitness comparison. In this experiment we compare the performance of each of the three proposed selections strategies in terms of maximum fitness value. Another measure of performance is the radius of the description. Given the same fitness value, the smaller the description the better the performance of



(a) (b)

Fig. 4. Fitness values and evolution with: (a) Population size 20. (b) Population size 100.



(a) (b)

Fig. 5. Comparison of different strategies with respect to (a) Fitness. (b) Radius.

the system. This implies if two different descriptors encompass the same number of sample points the smaller one should be preferred.

The comparison of different selection strategies with respect to the fitness of individuals is shown in Fig. 5(a). In this experiment the population size of 100 is used for all GAs. Among the three selection strategies, the $\mu\lambda$ selection converges faster to the target rate of 90% (dashed curve). The Rank based selection reaches the target rate in about 300 generations (solid curve). However, notice that the canonical GA fell in the local optima and never reached the target rate of 90% fitness (dotted curve).

Fig. 5(b) compares the description radius found by each of the three GAs. From the figure, the $\mu\lambda$ selection based GA converges to the actual solution faster than the other two methods. Rank-based selection finds the best solution in about 300 generations. However, after comparing the radius found by these methods it is clear that $\mu\lambda$ finds a better description than the rank-based selection. Notice that canonical GA was not able to find the optimal solution in the allowed number of generation and fell into one of the local optima.

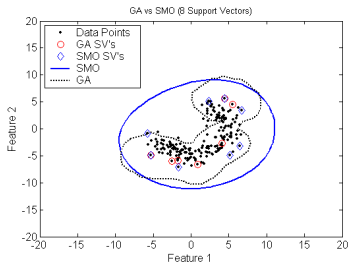


Fig. 6. Comparison of training the SVDD using the proposed GA and the SMO.

Table 1. Comparison of False Reject Rate for different classifiers.

Method	GA-SVDD	SMO-SVDD	MoG	KDE	KNN
FRR	0.0833	0.1067	0.1400	0.1667	0.1333

Comparison of the GA and the SMO. In this section we compare the results of training the SVDD by using the proposed GA and the SMO method.

The data set in this experiment is a banana shaped distribution of 200 data points in two dimensions. The distribution of data points can be seen in Fig. 6 along with the decision boundaries trained by using the GA and the SMO techniques. The GA is set to train the classifier and preserve only 8 support vectors. The SMO is used to achieve the same number of support vectors after training the SVDD. The classification boundary of the SVDD by using the proposed GA are represented by the dashed curve. The solid curve is the decision boundary for the SVDD trained by the SMO method. As seen, the GA gives better generalization compared to the SMO.

Quantitative comparison. We define the False Reject Rate (FRR) for a quantitative evaluation. By definition, FRR is the percentage of missed targets: $\text{FRR} = \frac{\#\text{Missed targets}}{\#\text{Samples}}$.

Table 1 shows a quantitative comparison between the proposed GA-based SVDD method and other techniques such as SVDD trained by SMO, Mixture of Gaussians (MoG), Kernel Density Estimation (KDE) and K-Nearest Neighbors (KNN). The FRR for SVDD is smaller than that of the other three which proves the superiority of this classifier for novelty detection.

5.2 Real videos

The foreground detection results of our method on real video sequences are shown and compared with existing statistical modeling techniques in this section.

Comparison in the presence of irregular motion. By using the *water surface* video sequence, we compare the results of foreground region detection using our proposed method with a typical KDE [5] and MoG [1]. Fig. 7(b), (c), and (d) show the results of the MoG, the KDE, and the proposed SVDD

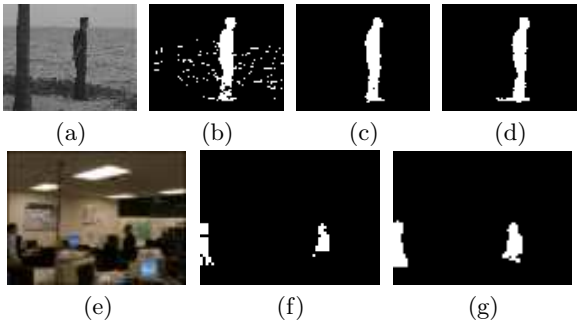


Fig. 7. Real videos: (a) *Water surface* sequence. (b) MoG. (c) KDE. (d) SVDD. (e) *Handshake* sequence. (f) KDE (g) SVDD.

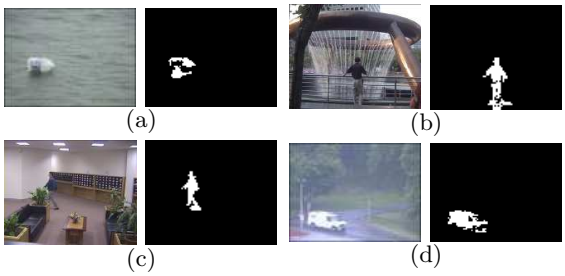


Fig. 8. Foreground detection results.

technique, respectively. As it can be seen, the proposed method gives a better detection.

Comparison in case of low contrast videos. Figure 7(e)-(g) shows the result of foreground detection using the proposed method in the *hand shake* video sequence and compares this result with that of the KDE method. As it can be seen from Fig. 7(f) and 7(g), the proposed method achieves better detection rates compared to the KDE technique.

Other difficult scenarios. Fig. 8(a)-(d) shows results of the proposed foreground detection algorithm in very difficult situations. Our system accurately detects the foreground regions in all of these situations robustly.

6 Conclusion and future work

Support Vector Data Description (SVDD) is an elegant technique to describe a single class of known samples, used in novelty detection. The main contribution of this work is to design an evolutionary computing algorithm (i.e. GA) to solve the optimization problem in training of the SVDD. The proposed SVDD is used in detecting foreground regions in videos with quasi-stationary backgrounds. The technique shows robust performance and fast convergence speed.

One future direction of this work is an incremental version of the proposed SVDD. This can be done by injecting new incoming data samples into the chromosome which replace the non-support vectors. The incremental SVDD will be adaptive to temporal changes in the sample set. The kernel and the classifier parameters can be encoded into the genetic materials as well. This unified framework evolves the data descriptor and the sample classifier together, thus, making the system automatic with respect to its parameters.

References

1. Stauffer, C., Grimson, W.: Learning patterns of activity using real-time tracking. IEEE Transactions on PAMI **22** (2000) 747–757
2. Lee, D.: Effective gaussian mixture learning for video background subtraction. IEEE Transactions on PAMI **27** (2005) 827–832
3. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.: Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. In proceedings of the IEEE **90** (2002) 1151–1163.
4. Mittal, A., Paragios, N.: Motion-based background subtraction using adaptive kernel density estimation. In proceedings of CVPR **2** (2004) 302–309
5. Tavakkoli, A., Nicolescu, M., Bebis, G.: Automatic robust background modeling using multivariate non-parametric kernel density estimation for visual surveillance. In proceedings of the International Symposium on Visual Computing **LNSC 3804** (2005) 363–370
6. Tavakkoli, A., Nicolescu, M., Bebis, G.: A novelty detection approach for foreground region detection in videos with quasi-stationary backgrounds. In proceedings of the 2nd International Symposium on Visual Computing (2006)
7. Tax, D., Duin, R.: Support vector data description. Machine Learning **54** (2004) 45–66
8. Sholkopf, B.: Support Vector Learning. Ph.D. Thesis, Technischen Universitat Berlin (1997)
9. Platt, J.: Sequential minimal optimization: A fast algorithm for training support vector machines. Microsoft Research Technical Report MSR-TR-98-14 (1998)
10. Lessmann, S., Stahlbock, R., Crone, S.F.: Genetic algorithms for support vector machine model selection. In proceedings of the International Joint Conference on Neural Networks (2006) 3063–3069
11. Liu, S., Jia, C.Y., Ma, H.: A new weighted support vector machine with ga-based parameter selection. In proceedings of the International Conference on Machine Learning and Cybernetics (2005) 4351–4355
12. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
13. DeJong, K.A.: Genetic algorithms are not function optimizers. Foundations of Genetic Algorithms (1993) 5–17