IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 5, Oct-Nov, 2013
ISSN: 2320 - 8791
www.ijreat.org

PD-OSVM: Pedestrian Detection Based on Online Support Vector Machine Using Convex Hull

RevathiM K[1], AnnapandiP[2] and Ramya K P[3]

[1]**Information Technology, Dr.Sivanthi Aditanar College of Engineering, Tiruchendur, Tamilnadu-628215, India**

[2]**Information Technology, Dr.Sivanthi Aditanar College of Engineering, Tiruchendur, Tamilnadu-628215, India**

[3]**Information Technology, Dr.Sivanthi Aditanar College of Engineering, Tiruchendur, Tamilnadu-628215, India**

Abstract

The support vector machine (SVM), an assuring new method for the classification, has been widely used in many areas efficiently. However, the online learning issue of SVM is still not addressed satisfactorily since when a new sample arrives to retrain the SVM to adjust the classifier. This may not be feasible for real-time applications due to the expensive computation cost for re-training the SVM. This paper propose an Online SVM classifier algorithm known as CH-OSVM, which is based on the convex hull vertices selection depends on geometrical features of SVM. From the theoretical point of view, the first d+1(d is the dimension of the input samples) selected samples are proved to be vertices of the convex hull. This guarantees that the selected samples in our method keep the greatest amount of information of the convex hull. From the pedestrian detection application point of view, the new algorithm can update the classifier without reducing its classification performance.

Keywords:*Kernel, online learning, machine learning, support vector machine, pedestrian detection, online classifier.*

1. Introduction

Support Vector Machines (SVMs) are a democratic machine learning method for classification, regression, and other learning tasks.Geometrically, it is to find a hyper plane with a maximal margin between two classes. The main advantages of the SVM method are summarized as follows. First, because of the combination of the empirical risk minimization and the prevention of over fitting, SVM can achieve a good generalization performance. Second, the problem of computing the hyper plane with a maximal margin can be converted into a convex quadratic optimization problem, which can be solved by quadratic programming techniques and has a global optimal solution. Finally, the obtained classifier is found out only by support vectors, and it can also be applied to nonlinear cases by using the kernel trick. Due to the above advantages, SVM has been successfully used in many real world problems:

- Hand-written characters can be recognized using SVMs.
- Pattern Recognition
- Hand-writing digital character recognition
- Face Recognition
- Classification tasks of text

However, a main drawback of SVM is that it requires a long time for training and a lot of memory for dealing with large scale data sets. Online learning has significant applications in real-time pattern recognition systems, such as pedestrian detection and aircraft visual navigation systems. In order to address the online learning issue of SVM, several successful algorithms has been proposed, which will be reviewed in the next section. There are few works that consider updating SVM online by preserving the skeleton samples based on the geometric characteristics of SVM. Geometrically, for linearly separable data sets, the computation of SVM is equivalent to the problem of computing the nearest points between the convex hulls formed by the positive and negative samples [7]-[9]. Therefore, we can permanently and safely delete samples within the convex hulls and only preserve the vertices of the convex hulls for online learning. The classification result of the online classifier trained with only the vertices of the convex hulls and newly arrived samples is the same as that obtained by retraining with all samples. Based on the above motivation, in this paper, we propose an online SVM classification

algorithm. A main advantage of our approach over the existing ones is that it can deal with large-scale data sets efficiently since only a small number of samples are selected for online learning. Based on the above motivation, in this paper, we propose an online SVM classification algorithm. A main advantage of our approach over the existing ones is that it can deal with large-scale data sets expeditiously since only a small number of samples are selected for online learning. There are two main steps in our algorithm, which are described as follows.

1) Offline Process: The skeleton vertices of convex hulls, which are formed by the current positive and negative training samples, are selected and stored for the following online process.

2) Online Process: The SVM classifier is updated based on samples selected in the off-line process and the newly arriving samples whose distances to the current classification hyperplane are within a given threshold.

We apply the offline process only in the case when the number of the current training samples, which consist of the selected samples and all newly arrived samples exceeds a given threshold. We execute sporadically the off-line step in order to reduce the number of the current training samples. This makes the online learning with newly arrived samples possible.

2. Existing Online SVM Algorithms

In this section, the existing online learning methods for SVM are briefly reviewed. Singh *et al.* [41] proposed a method called 2*v*-OGSSVM, in which we first construct the decision hyperplane using an initial training dataset and then retrain the classifier by incorporating the new training data points into the decision hyperplane. It also removes unnecessary and irrelevant data so that the number of support vectors does not increase drastically with the increase in training samples. Thus, the online learning algorithm includes both incremental and decremental learning. Bordes *et al.* [42] proposed an online method called LASVM, which is based on SMO (Sequential minimal optimization). When new samples arrives update the classifier is based on the alternating two kinds of direction searches, called PROCESS and REPROCESS. Cheng *et al.* [44] proposed an incremental SVM by using active query. Instead of selecting training samples randomly, we divide them into groups and apply the k-means clustering algorithm to collect the initial set of training samples. In active query, we assign a weight to each sample according to its confidence factor and its distance to the separating hyperplane. The classifier is updated based on the training samples selected by active query, and the iteration stops when there are no unused informative training samples. Lau *et al.* [45] presented an online support vector classifier, in which the classifier is adjusted by retraining the SVM with the current support vectors, the samples violating the Karush-Kuhn-Tucker (KKT) conditions and the newly arriving samples. The updating process stops only if no sample violates the KKT conditions and no new sample arrives. Bordes *et al.* [48] proposed an efficient online SVM algorithm, named Huller. When a new misclassified sample arrives, Huller attempts to make this new sample become a support vector and removes another support vector from the current classifier via the update of the nearest points of the convex hulls. The classifier is thus adjusted. The Huller algorithm is closely related to our method in this paper. Both our algorithm and the Huller algorithm update the classifier based on the geometrical interpretation of SVM. However, the numerical objective of the two algorithms is different. Our method tries to select the vertices of the convex hulls of the current training samples in the off-line step and then retrains the classifier using these selected samples and newly arrived samples. Since the training set is greatly reduced in the off-line step, the online learning is tractable. In each online iteration, our algorithm gives the optimal solution of the SVM classifier trained with the selected samples and all newly arrived samples, while, when a new sample arrives, the Huller algorithm tries to decrease the distance of the current nearest points to update the classifier. But the classifier obtained by Huller is not the optimal solution of the SVM classifier in the current iteration.

3. CH Vertices Selection Algorithm

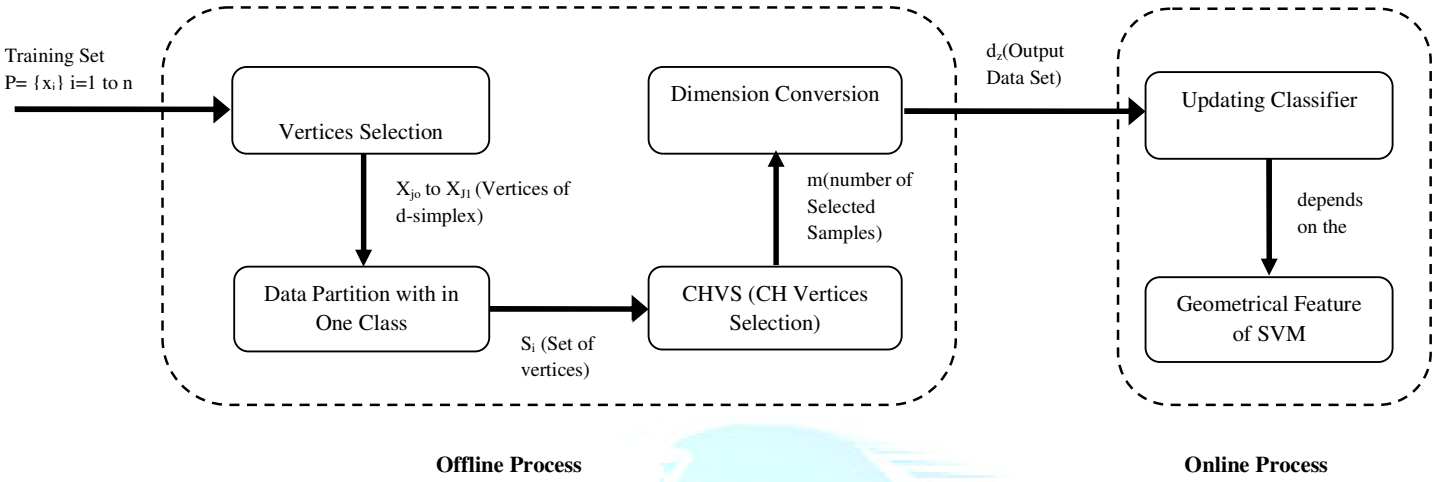

Fig. 1- Overview of CH-OSVM Method

In this section, we give a detailed description of the proposed method. In Section 3-A, we introduce the kernel convex hull distance. In Section 3-B, we use a synthetic example to illustrate the proposed method.

A.Kernel Convex Hull Distance

The kernel convex hull of the set $P = \{x_i\}_{i=1}^n \subset \mathbb{R}^d$ is defined as:
$con(P) = \{\sum_{i=1}^n \alpha_i x_i | x_i \in P, \sum_{i=1}^n \alpha_i = 1, \alpha_i \geq 0, i = 1, \ldots, n\}$.

Given a set $P = \{x_i\}_{i=1}^n \subset \mathbb{R}^d$ and a point $x \in \mathbb{R}^d$, the Euclidean distance between and x and $con(P)$ can be computed by solving the following quadratic optimization:

$$\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - c^T\alpha$$

$$s.t.\ e^T\alpha = 1, \alpha \geq 0 \qquad (1)$$

Where $e = [1,1,\ldots,1]^T, Q = X^T X and c = X^T x$ with $X = [x_1, \ldots, x_n]$. suppose the optimal solution of (1) is α^*.then the convex hull distance between x and $con(P)$ is given by

$$d_c(x, con(P)) = \sqrt{x^T x - 2c^T\alpha^* + \alpha^{*T} Q\alpha^*}. \quad (2)$$

B. Synthetic Example

Before formally introducing the samples selection method (offline process), we present a synthetic example in R2 (consider Fig. 2) to explain the basic idea of the proposed method. As shown in Fig. 2, we first select three samples with the following steps. First, randomly select a sample, denoted by x0. Second, the furthest sample to x0 is selected and denoted by x1.

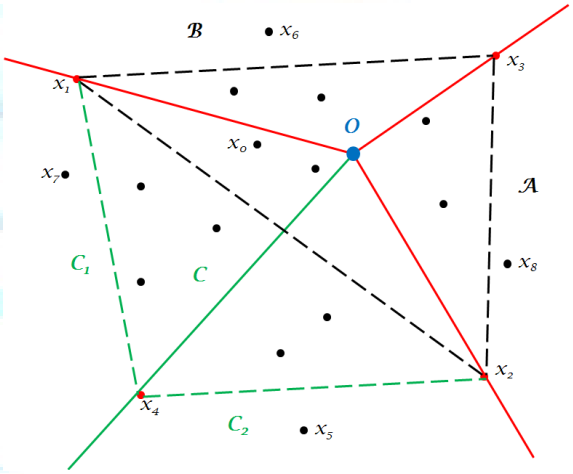

Fig. 2- Conventional diagram of CH vertices selection algorithm

Thirdly, the furthest sample to x1 is selected and denoted by x2. Finally, the sample that is the furthest to the straight line passing through x1 and x2 is selected and denoted by x3. To select the d + 1 samples that can construct an as large d-simplex as possible. Suppose O is the mean of the samples $\{x1, x2, x3\}$. Then radials Ox1, Ox2 and Ox3 divide the samples outside of $con(\{x1, x2, x3\})$ into three parts, denoted by A, B, and C, severally. From Fig. 2, it can be seen that the number of samples in part C is still la e, so we need to further divide this part into several smaller parts. In part C, the sample which is furthest to the line x1x2 is selected and denoted by x4. Then the radial Ox4 further divides the samples in C but outside of $con(\{x1, x2, x3, x4\})$ into two parts, denoted by C1 and C2. So far, we have divided the samples outside of $con(\{x1, x2, x3, x4\})$into four parts, and the numbers of samples in the four parts are all small. The edges $x2x3, x1x3, x1x4, x2x4$are called the corresponding edges of parts A, B, C1, and C2, respectively. From

Fig. 1, it can be seen that the distance to $con(\{x1, x2, x3, and x4\})$ of each sample in each part is equal to its distance to the representing edge of that part. For example, the distance to $con(\{x1, x2, x3, x4\})$ of each sample in part A is equal to the distance of the sample to $con(\{x2, x3\})$ (the corresponding edge of part A). We call the convex hull of the samples in one part as a sub-convex hull which is initialized to be the representing edge of that part, that is, the sub-convex hulls of parts A, B, C1, and C2 are initialized to be the edges $x2x3, x1x3, x1x4, x2x4$, respectively. We individually compute the distances between the samples and the sub-convex hull in each part. Then the furthest sample is selected, which is denoted as x5. The sub-convex hull of part C2 which x5 belongs to is then spanned by adding x5 to the vertices of the sub-convex hull. The distances between the samples in part C2 and the spanned sub-convex hull are recomputed, while the distances between samples in other parts and their corresponding sub-convex hulls will not be changed. Then the next furthest sample will be selected. In such a procedure, samples $x6, x7, and x8$ are selected in turn. Note that if we denote by m_d the distance between the next chosen sample and the sub-convex hull of the part which it belongs to, then the convex hull of all selected samples can approach any sample with an error not exceeding m_d.

4. Online SVM based CH-VS Algorithm

In Fig. 3, Ni s+ and Ni s− represent the numbers of samples that we need to select from the positive samples set P+ and the negative samples set P−, respectively, by using the CHVS algorithm (see the fourth box) in the i^{th} iteration. Train($P_s^{(i)}$) represents the fact that the sample set $P_s^{(i)}$, which is selected in the i^{th} iteration, is used to train the SVM classifier. The classifier is denoted by r_i, and $a_i$is the classification rate of r_i on the testing samples.

$a_\delta = [a_\delta j]_{1\times 10}$is an increment vector of the classification rate, where$a_\delta^j = a_{i-10+j} - a_{i-10}$. $a_\delta \leq \varepsilon$ represents that all elements of $a_\delta$are not larger than ε, where ε is a small threshold. This means that the set $P_s^{(i-10)}$of selected samples already has thegreatest amount of information of the training samples. In ourexperiments, we always set $\varepsilon = 1/|P_T|$, where $|P_T|$is thenumber of the testing samples. As shown in Fig. 3, in each iteration, we add one percent of the training samples on the basis of the selected samples set in the last iteration. That is, in the *i*-iteration of VS algorithm, we select$[N_+ * i * 0.01]$positive samples and $[N_- * i * 0.01]$negative samples, respectively, by applying the CHVSalgorithm to each class. We use these selected samples to train the SVM classifier. Generally speaking, the classification rate on testing samples increases with the number of selected samples. However, when the selected samples keep the greatest amount information of the convex hull, the classification rate will either not increase or only increase a little with the number of selected samples further increasing. This is the reason why we use the increment of the classification rate on testing samples as the stop criterion in Fig. 3. We do not further select samples if the increment of the

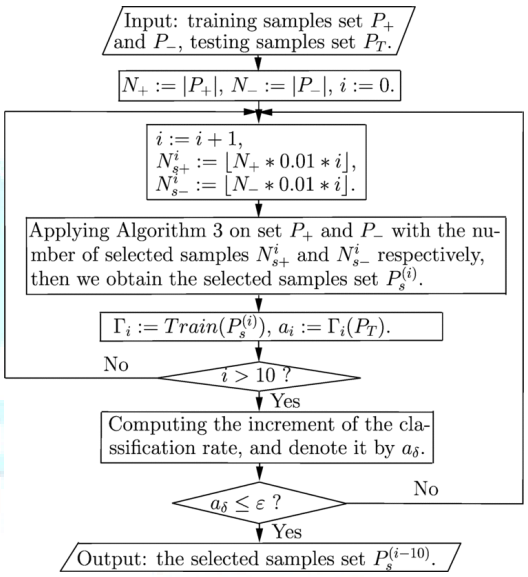

Figure 3 CH-VS Algorithm

classification rate within a small threshold ε. Hence, the number *m* of the selected samples in each class is adaptively determinedvia the increment of the classification rate on the testingsamples.

5. Expected Experimental Results

The classification rate of the four methods on the testing samples is shown in Table II. It can be seen that our method gives almost the same classification results as LIBSVM and outperforms RS-OSVM and LASVM (except for Cod-rna). This is because: 1) in the off-line step, we select the skeleton vertex samples which keep the greatest amount of information of the current training samples, and 2) in the process of the online step, we consider not only the new samples being closed to the current classifier margin, but also all the samples which have been involved in the previous updating process (these samples not only contain support vectors of the previous classifiers, support vectors and some non-support vectors of the current classifier and all the skeleton vertex samples selected in the off-line step, but will also become support vectors at the end with a high probability). Thus, our method rarely deletes important samples in the training process and retains the good generalization power of LIBSVM trained with all the samples. LASVM with online setup

is only based on the current classifier to cache samples. Once the samples are kicked out from the cache in the REPROCESS step, they are discarded forever. LASVM with online setup would discard some non-negligible support vectors of the final true classifier in the REPROCESS step, and therefore its classification rate is lower than that of our method. On the other hand, it should be pointed out that the nine data sets are linearly or nonlinearly separable and therefore, by using the kernel trick, the classification rate of our method and LIBSVM on these data sets is almost the same.

Table 1- Classification Rate of LIBSVM, LASVM, RS-OSVM, AND VS-OSVM

Datasets	LIBSVM	LASVM	RS-OSVM	CH-OSVM
UCI-Stalog	99.259±0.151	98.750±0.151	98.673±0.220	99.275±0.170
UCI-Letter	99.893±0.107	99.427±0.150	98.549±0.937	99.893±0.107

6. Conclusion

This paper provides a new online SVM classifier based on the selection of vertices of the convex hull in each class. In the samples selection process, a small number of skeleton samples, which constitute the approximate convex hull of the current training samples, were selected for online learning. In the online classification process, the classifier was updated online based on the selected samples and newly arriving samples. It was proved theoretically that the $d + 1$ selected samples are all vertices of the convex hull. It guarantees that the samples selected by our approach retain the greatest amount of information of the current training samples. From the application point of view, the online adjustment of an SVM classifier based on the selected skeleton samples improves the classification rate very little but consumes much less time. Therefore, the proposed online SVM can be applied to various online classification tasks, such as pedestrian detection and visual tracking. Experimental results on real-world data sets have validated the effectiveness of the proposed method. We should remark that our method may not be directly applied to data sets with heavy noise since it is based on the vertices selection of the convex hull of the samples in each class. For heavily noised data sets, the involved training samples can be preprocessed by using de-noising methods before samples selection in the off-line step, that is, we can add a preprocessing step at the beginning of Step 2 in the VS-OSVM algorithm. Recently, Geebelen*et. al.* [16] proposed a method for pruning the solutions of the SVM, which first removes the misclassified samples or flips their labels to make the training set separable and then formulates an approximate SVM solution with the upper bounded weight vector for the modified separable training set. Inspired by this method, we can also remove the current misclassified samples or swap their labels before the samples selection step to reduce the heavy noise. However, we cannot eliminate the newly arrived noised data samples during the online step; these samples may cause the updated classifier to have some deviation from the final true classifier. How to eliminate the heavily noised data samples in the online step is still an intractable issue, and it will be one of our future works. In Step 3 of the VS-OSVM algorithm, the classifier is updated only when there are newly arrived samples whose distances to the current classifier are within a given threshold, since we believed that these samples would become support vectors at the end with a very high probability. Hence, the initial data set is important for the training process of our classifier. If the initial data set cannot indicate the distribution of the whole data set, then the initial classifier trained with the selected samples may result in a large deviation from the final true classifier. Then some important support vectors of the final true classifier will be possibly deleted in the online updating process, and the generalization performance of our classifier will be reduced. Nevertheless, in each of our experiments, we randomly selected a half of the data samples as the initial data set, which can indicate the distribution of the whole data set to a great extent. Therefore, we obtained good experimental results. How to find a more effective method of selecting the initial data set may be an interesting issue, and it will be our future work.

Reference

[1]J. An, Z. Wang, and Z. Ma, "An incremental learning algorithm forsupport vector machine," in *Proc. Int. Conf. Mach. Learn. Cybern.*,Nov. 2003, pp. 1153–1156.

[2] R. Singh, M. Vatsa, A. Ross, and A. Noore, "Biometric classifier updateusing online learning: A case study in near infrared face verification,"*Image Vis. Comput.*, vol. 28, no. 7, pp. 1098–1105, Jul. 2010.

[3] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, "Fast kernel classifierswith online and active learning," *J. Mach. Learn. Res.*, vol. 6,pp. 1579–1619, Sep. 2005.

[4] M.Wang, X. Zhou, F. Li, J. Huckins, R. King, and S. T. C.Wong, "Novelcell segmentation and online SVM for cell cycle phase identification inautomated microscopy," *Bioinformatics*, vol. 24, no. 1, pp. 94–101, 2008.

[5] S. Cheng and F. Shih, "An improved incremental training algorithm forsupport vector machines using active query," *Pattern Recognit.*, vol. 40,no. 3, pp. 964–971, 2007.

[6] K. Lau and Q. Wu, "Online training of support vector classifier," *PatternRecognit.*, vol. 36, no. 8, pp. 1913–1920, 2003.

[7] G. Cauwenberghs and T. Poggio, "Incremental and decremental supportvector machine learning," in *Advances in Neural Information Processing Systems*, Cambridge, MA: MIT Press, 2001, pp. 409–415.

[8] D. Wang, B. Zhang, P. Zhang, and H. Qiao, "An online core vectormachine with adaptive MEB adjustment," *Pattern Recognit.*, vol. 43,no. 10, pp. 3468–3482, 2010.

[9] A. Bordes and L. Bottou, "The Huller: A simple and efficient onlineSVM," in *Proc. 16th Eur. Conf. Mach. Learn.*, Oct. 2005, pp. 505–512.

[10] J. X. Dong, A. Krzyzak, and C. Y. Suen, "A practical SMO algorithm,"in *Proc. Int. Conf. Pattern Recognit.*, vol. 3. Aug. 2002, pp. 1–4.

[11] X. F. He and P. Niyogi, "Locality preserving projections," in *Advancesin Neural Information Processing Systems*. Cambridge, MA: MIT Press,2003.

[12] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, "Face recognition usingLaplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3,pp. 328–340, Mar. 2005.

[13] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction bylocally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326,Dec. 2000.

[14] L. K. Saul and S. T. Roweis, "Think globally, fit locally: Unsupervisedlearning of low dimensional manifold," *J. Mach. Learn. Res.*, vol. 4,pp. 119–155, Jan. 2003.

[15] H. Qiao, P. Zhang, D. Wang, and B. Zhang, "An explicit nonlinearmapping for manifold learning," *IEEE Trans. Cybern.*, vol. 43, no. 1,pp. 51–63, Feb. 2013.

[16] F. Orabona, C. Castellini, B. Caputo, L. Jie, and G. Sandini, "Onlineindependent support vector machines," *Pattern Recognit.*, vol. 43, no. 4,pp. 1402–1412, Apr. 2010.