

Adapting SVM Image Classifiers to Changes in Imaging Conditions Using Incremental SVM: An Application to Car Detection

Epifanio Bagarinao¹, Takio Kurita¹, Masakatsu Higashikubo², Hiroaki Inayoshi¹

¹Neuroscience Research Institute

National Institute of Advanced Industrial Science and Technology

Tsukuba City, Ibaraki, 305-8568 Japan

{epifanio.bagarinao, takio-kurita, h.inayoshi }@aist.go.jp

²Sumitomo Electric Industries, Ltd.

Shimaya, Konohana-ku, Osaka, 554-0024 Japan

higashikubo@sei.co.jp

Abstract. In image classification problems, changes in imaging conditions such as lighting, camera position, etc. can strongly affect the performance of trained support vector machine (SVM) classifiers. For instance, SVMs trained using images obtained during daylight can perform poorly when used to classify images taken at night. In this paper, we investigate the use of incremental learning to efficiently adapt SVMs to classify the same class of images taken under different imaging conditions. A two-stage algorithm to adapt SVM classifiers was developed and applied to the car detection problem when imaging conditions changed such as changes in camera location and for the classification of car images obtained during day and night times. A significant improvement in the classification performance was achieved with re-trained SVMs as compared to that of the original SVMs without adaptation.

Keywords: incremental SVM, car detection, constraint training, incremental re-training, transfer learning

1 Introduction

The effective training of support vector machine (SVM) usually requires a large pool of training datasets. However, gathering datasets for SVM learning takes longer times and needs more resources. Once trained, SVM classifiers cannot be easily applied to new datasets obtained from different conditions, although of the same subject. For instance, in image classification problems, changes in imaging condition such as lighting, camera position, among others, can strongly affect the classification performance of trained SVMs making the deployment of these classifiers more challenging.

Consider for example the detection of cars in images from cameras stationed along highways or roads as a component of an intelligent traffic system (ITS). The problem is to detect the presence of cars in sub-regions within the camera's field of view. To solve this problem, one can start collecting images from a given camera, extract

training datasets from these images, and train an SVM for the detection problem. After training, the SVM could work perfectly well for images obtained from this camera. However, when used with images taken from other cameras, the trained classifier could perform poorly because of the differences in imaging conditions. The same can be said of SVMs trained using images obtained during daylight and applied to classify images taken at night.

One solution to this problem is to train SVMs for each camera or imaging condition. But this can be very costly, requires significant resources, and takes a longer time. An ideal solution is therefore to be able to use an existing large collection of training datasets to initially train an SVM and adapt this SVM to the new conditions using a minimal number of additional training sets. This involves transferring knowledge learned from the initial training set to the new conditions.

This problem is related to the topic of transfer learning (for instance [1-3]). Closely related to this work is that of Dai and colleagues [3]. They presented a novel transfer learning framework allowing users to employ a limited number of newly labeled data and a large amount of old data to construct a high-quality classification model even if the number of the new data is not sufficient to train a model alone. Wu and Dietterich [4] also suggested the use of auxiliary data sources, which can be plentiful but of lower quality, to improve SVM accuracy. The use of unlabeled data to improve the performance on supervised learning tasks has also been proposed by Raina, et al.[5].

In this paper, we investigate the use of incremental SVM [6-7] to improve the classification of the same class of images taken under different imaging conditions. We assumed the existence of a large collection of labeled images coming from a single camera that could be used as starting training samples. Two training methods based on incremental SVM are employed for the initial training. One is the standard incremental approach, henceforth referred to as *non-constraint training*. The other one is *constraint training*, which imposes some limitations on the accepted support vectors during the learning process. After training, SVMs are adapted by means of incremental re-training to the new imaging condition using only a small number of new images. This is the transfer learning stage. The algorithms used will be detailed in the next section. In our experiments, we used images captured from cameras stationed along major roads and highways in Japan. Sub-images containing cars or background (road) were extracted and their histogram of oriented gradient (HOG) [3] computed. HOG features were then used as training vectors for SVM learning.

2 Materials and Method

In this section, we first give a brief discussion of the standard incremental SVM approach (non-constraint training). The constraint training method is then presented in the next subsection and finally incremental re-training is discussed.

2.1 Incremental SVM

Incremental SVM [6-7] learning solves the optimization problem using one training vector at a time, as opposed to the batch mode where all the training vectors are used

at once. Several methods had been proposed, but these mostly provided only approximate solutions [9-10]. In 2001, an exact solution to incremental SVM was proposed by Cauwenberghs and Poggio (CP) [6]. In the CP algorithm, the Kuhn-Tucker (KT) conditions on all previously seen training vectors are preserved while “adiabatically” adding a new vector to the solution set.

To see this, let $f(\mathbf{x}) = \sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$ represents the optimal separating function with training vectors \mathbf{x}_i and corresponding labels $y_i = \pm 1$. The KT conditions can be written as (see [1] for more details):

$$g_i = f(\mathbf{x}_i) y_i - 1 = \begin{cases} \geq 0, & \alpha_i = 0 \\ = 0, & 0 < \alpha_i < C, \\ \leq 0, & \alpha_i = C \end{cases}, \quad (1)$$

$$h = \sum_{i=1}^n \alpha_i y_i \equiv 0, \quad (2)$$

with C being the regularization parameter, the α_i 's are the expansion coefficients, and b the offset. The above equations effectively divides the training vectors into three groups, namely, margin support vectors ($g_i = 0$), error support vectors ($g_i < 0$), and non-support vectors ($g_i > 0$).

In the CP algorithm, a new training vector is incorporated into the solution set by first setting its α -value to 0, and then its g -value is computed using Eq. (1). If it is greater than 0, then the new training vector is a non-support vector and no further processing is necessary. If not, the training vector is either an error vector or a support vector and the initial assumption that its α -value is 0 is not valid. The α -value is then recursively adjusted to its final value while at the same time preserving the KT conditions on all existing vectors. For a detailed discussion, refer to Ref. [6]. In this paper, the use of the original CP algorithm for training is referred to as non-constraint training as compared to constraint training, which will be discussed next. We also used an in-house implementation of this algorithm using C to support incremental SVM learning.

2.2 Constraint Training

Supposed we have an initial training set labeled as $\mathbf{X} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$ where n is significantly large. In our problem, the training vector \mathbf{x}_i represents the HOG features from images coming from a given camera, while the label y_i can either be ‘with car’ (1) or ‘no car’ (0). Also, let $\mathbf{Z} = \{(\mathbf{z}_1, y_1), (\mathbf{z}_2, y_2), \dots, (\mathbf{z}_m, y_m)\}$ denotes another dataset taken from another camera and $m \ll n$. We defined constraint training as the selection of an appropriate set of support vectors from \mathbf{X} that maximizes the classification performance of an SVM using dataset \mathbf{Z} as test set. It should be noted that the support vectors of \mathbf{X} do not necessarily give an optimal classification of \mathbf{Z} as will be shown in the results. However, there may exist a subset of \mathbf{X} with support

vectors that can lead to an optimal classification of \mathbf{Z} . Finding this subset is the aimed of constraint training.

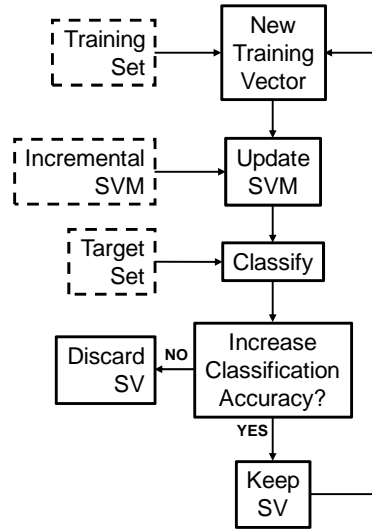


Figure 1. Incremental SVM-based constraint training: 1) Get a new training vector from the training set \mathbf{X} . 2) Update existing SVM using incremental approach. 3) Test updated SVM using target dataset \mathbf{Z} . 4) If classification accuracy increases, keep the support vector (SV); otherwise, discard it. 5) Repeat (1) until all training vectors are processed

There are several ways to implement constraint training. An example is to randomly extract subsets from \mathbf{X} , train SVMs using these subsets, and select the subset which gives the maximum classification accuracy of \mathbf{Z} . In this paper, we used incremental SVM and employed the algorithm outlined in Fig. 1. The algorithm follows from that of the non-constraint case except for an additional constraint, which is given as follows. As each new training vector is added, a classification test using a target dataset (\mathbf{Z} in the notation above) is performed. A newly computed support vector is included into the running solution set if it increases the classification accuracy of the evolving SVM; otherwise, support vectors that tend to decrease the classification accuracy are discarded. The algorithm ensures that only support vectors that increase the classification accuracy of the target set are included in the final SVM. This effectively realigns the separating function of \mathbf{X} to give an optimal separation of samples from \mathbf{Z} , without using any sample from the latter. The final set of support vectors is the desired subset of \mathbf{X} .

2.3 Incremental Re-training

In constraint training, only samples from the initial training set \mathbf{X} are used during training. Samples from the target set \mathbf{Z} are not included in the training process. In incremental re-training, the trained SVM is updated using this limited number of

samples. See Fig22. 2. The basic idea is that an SVM is already trained using dataset \mathbf{X} via either non-constraint or constraint training method. After this initial training, samples from dataset \mathbf{Z} are incrementally incorporated into the trained SVM to adapt it to the new dataset. This is the transfer learning phase. The final result is an adapted SVM with a much improved classification performance of the class of feature vectors represented by the target dataset \mathbf{Z} .

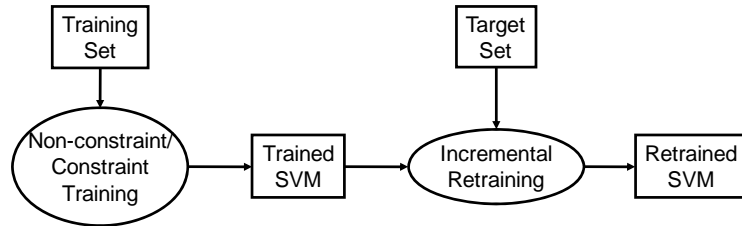


Figure 2. With incremental re-training, an SVM is trained using an initial training set either via non-constraint or constraint approach, then is re-trained using the target set by means of incremental SVM

3 Results

Five cameras stationed at five different locations along major roads in Japan were used to capture images of passing vehicles. For each camera image, 16×16 sub-images containing cars and no cars were extracted. These selected images were then converted into HOG features using an 8×8 overlapping blocks with 4 pixels overlap giving a total of 9 blocks. The direction of gradient in each block was divided into 8 effectively converting each 16×16 image into a 72-dimensional feature vector.

From this, five groups of datasets were formed and labeled as follows: 379LCR (58,453 feature vectors), 382LR (56,405 feature vectors), 383LR (50,058 feature vectors), 122LR (12,762 feature vectors), and 384LCR (61,214 feature vectors). Images from 379LCR, 382LR, and 383LR were taken during daytime, while that from 122LR and 384LCR were obtained at night. Each dataset was then divided into 10 subsets. For each subset, an SVM was trained and then tested using the other subsets. The optimal SVM kernel (selected from linear, polynomial, RBF, and sigmoid kernels) and the associated kernel parameters were chosen using a cross validation approach with a grid search method in the parameter space. Each subset can have different optimal kernel and kernel parameters.

In our first experiment, we looked into the classification performance of SVMs trained using the non-constraint approach. The within group classification performance where SVMs were trained and tested using datasets belonging to the same group was evaluated. For instance, an SVM trained using 379LCR_0 was used to classify 379LCR_n, where $n = 1, \dots, 9$. We also tested cross group classification performance where SVMs trained from one group were used to classify data from another group.

The classification performance for different SVMs trained using subsets of 384LCR are shown in Table 1. The first column indicates the subset used to train the SVM, while the rest of the columns show the classification accuracy. The second column is the result for within group classification, while columns 3 to 6 are the results for cross group classification. For within group classification, the classification accuracy is significantly high (more than 99%). The same performance can be said for the other datasets not shown. On the other hand, the performance for cross group classification varies depending on the dataset used as test set. For some dataset, the performance can be as high as 99% (e.g., table 1, column 6), while for others, it can be lower than 70% (e.g., table 1, column 4). 384LCR-based SVMs are poor classifiers for 382LR or 383LR (both daytime datasets), but performed relatively well in classifying 379LCR (daytime dataset) or 122LR (nighttime datasets).

Table 1. Classification accuracy (%) of SVMs trained using subsets of 384LCR

384LCR_n	384LCR	379LCR	382LR	383LR	122LR
0	99.9739	84.6013	75.1565	74.8072	98.5896
1	99.9804	90.4915	67.9231	75.1508	99.8433
2	99.9869	91.2442	72.4156	79.7095	99.8668
3	99.9902	88.5310	69.6002	77.0167	99.8198
4	99.9755	91.1023	73.3942	80.6065	99.7649
5	99.9739	90.5240	73.3907	80.7084	99.7649
6	99.9771	91.4256	72.7755	80.2269	99.7963
7	99.9820	91.2152	80.9450	86.7913	99.8041
8	99.9902	90.1887	70.3555	77.6399	99.8590
9	99.9788	91.6240	77.9559	83.3593	99.8746

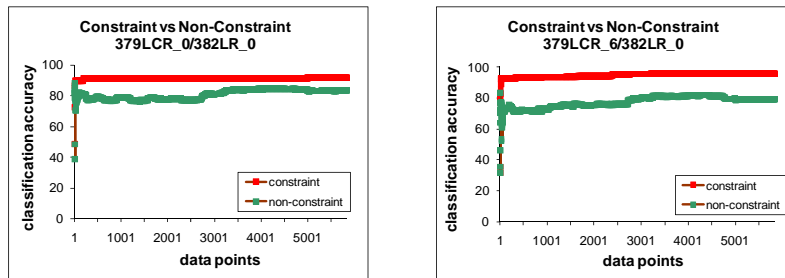


Figure 3. Classification accuracy during constraint (red plots) and non-constraint (green plots) training as the number of incorporated samples is increased. In both panels, 382LR_0 was used as the target/test set, while 379LCR_0 (left) and 379LCR_6 (right) as the training sets

For the next experiment, we examined whether it is possible to improve cross group classification accuracy using constraint training. We also compared the

accuracy of the constraint method with that of the non-constraint method for cross group classification during incremental learning. The results are shown in Fig. 33. For non-constraint training, the classification accuracy of the target set fluctuates as the number of support vectors increases. This is shown in the green plots of both panels. Since constraint training considers only support vectors that can increase the classification accuracy of the target set, the plots shown in red are always increasing. Interestingly, the classification accuracy with constraint training exceeded that of the non-constraint case even with this simple condition. By imposing this constraint to the learning process, the performance of the evolving SVM as a classifier of the target set is considerably improved.

Table 2. Classification accuracy (%) of SVMs trained with constraint and using subsets of 379LCR as training sets and 382LR_0 as target set

379LCR_n	379LCR	382LR	383LR	384LCR	122LR
0	91.2186	91.0664	93.3217	90.6786	93.3004
1	86.4113	94.7753	93.5515	95.9650	97.7198
2	86.3121	93.6920	92.0253	92.6275	96.4739
3	86.3172	93.9863	92.7764	97.9841	98.8011
4	79.6931	93.3215	92.1611	94.1419	96.3329
5	78.2252	92.8127	92.7164	92.4756	95.2359
6	87.0819	95.2256	93.7712	92.8742	96.4582
7	83.0770	93.3481	93.4796	96.1463	96.0586
8	86.0674	88.6038	91.3760	98.4415	97.4220
9	88.8885	95.1299	94.4924	97.0056	98.0959

Table 3. Classification accuracy (%) results for incremental re-training. Subsets of 379LCR were used as training sets and 384LR_0 as the target set

379LCR_n	Incremental Retraining			
	Constraint		Non-constraint	
n	379LCR	384LCR	379LCR	384LCR
0	89.0613	99.9951	99.6750	99.9935
1	90.9996	99.9886	99.7109	99.9935
2	93.0765	99.9935	99.6613	99.9902
3	96.0669	99.9706	99.7006	99.9951
4	90.9329	99.9967	99.7793	99.9967
5	90.6660	99.9967	99.8084	99.9918
6	82.0351	98.6686	99.7536	99.9886
7	92.5308	99.9984	98.8777	99.4527
8	90.9466	99.9951	99.7827	99.9918
9	97.9693	99.9918	99.6921	99.9820

Table 2 shows the classification performance of the trained SVMs with constraint. Two effects can be observed. As expected, the cross group classification accuracy generally increases for all datasets considered. This is particularly true for the target dataset, which in this case is 382LR, with more or less 10% increase in accuracy (column 3). On the other hand, a corresponding decrease in the within group classification accuracy can also be observed (column 2). But this is just expected since constraint training is designed to optimize the classification of the target set, which may not result in an optimal solution for the original training set.

We next used incremental re-training to include the limited number of target dataset into the training process and evaluated the classification performance of the resulting SVMs. Both the use of non-constraint and constraint approaches for initial training was investigated. The results are summarized in Table 333. Here, subsets of 379LCR were used as initial training sets and the resulting SVMs were incrementally re-trained using 384LR_0. With incremental re-training, the classification accuracy of the target dataset (384LCR) jumps to more than 99%, a significant improvement compared to the cross group classification performance. Moreover, for non-constraint initial training, the classification accuracy of the initial training dataset (379LCR) remains high.

To evaluate the number of additional training vectors needed to raise the classification accuracy during retraining, we performed classification test for each newly added support vectors as the SVM evolved. We randomly shuffled the order the additional training vectors are incorporated into the training process and took the average of the classification result. Representative plots are shown in Fig. 4. The left panel shows the classification performance of an SVM initially trained using 379LCR_0, and then incrementally re-trained using 382LR_0. On the other hand, 384LCR_0 was used for re-training in the right panel. Both panels showed that the constraint approach for initial training (blue plots) made the trained SVMs adapt faster (less additional training samples) to the new datasets as compared to the use of non-constraint approach for initial training. In both cases, only a small number of target vectors are needed to adapt the SVMs to the new datasets.

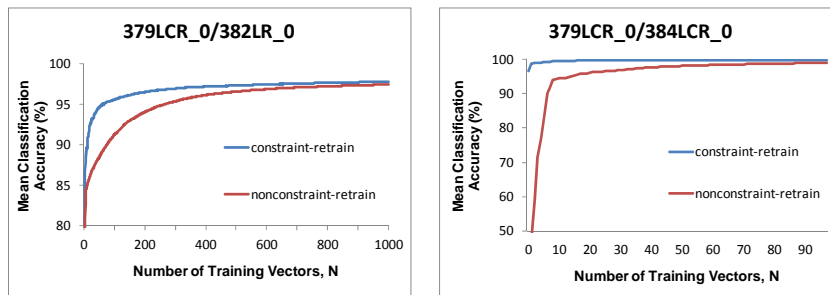


Figure 4. Classification accuracy as a function of the number of target samples incorporated into the incremental re-learning process. The SVMs were initially trained using 379LCR_0 and then incrementally retrained using 382LR_0 (left panel) and 384LCR_0 (right panel). Both constraint (blue plots) and non-constraint (red plots) approaches for initial training were evaluated

4 Discussion

Support vector machines are robust classifiers for datasets they are initially trained. Changes to some of the conditions where the initial datasets were acquired could significantly affect the classifiers' performance. The case currently considered is the detection of cars in images from cameras stationed along major roads. From the results presented, differences in lighting conditions from one location to the other can significantly affect the classification performance of SVMs trained using only dataset from one location (see Table 1). This presents a significant problem in the deployment of these classifiers. Training an SVM for each location can mitigate this limitation for small deployment. But for large scale application, this can be very costly, requires significant resources, and takes a longer time, and therefore, impractical.

In this paper, we have demonstrated a practical approach to overcome this limitation. The approach requires that an initial collection of dataset, possibly from a single camera, be available. For new deployment, a small number of additional images can be taken, which can then be used to adapt an existing classifier to the new imaging condition via transfer learning. Since the approach is based on incremental SVM, it is also possible to do on-line learning. The general idea is that from the initial collection, a subset can be selected that optimizes the classification accuracy of the new dataset using constraint training. The resulting SVM can then be incrementally re-trained using the additional (target) dataset to further improve its classification performance.

The result in Table 2 shows the efficacy of the constraint approach to extract subsets from the initial training set which can maximize the classification accuracy of the target set. Although the classification accuracy of the initial dataset decreases, this is immaterial since the final goal is the improvement in the classification of the new dataset for deployment purposes. The extracted subsets can then be combined with the target set via incremental re-training to further improve the classification of the target set as shown in Table 33. This has the advantage of faster adaptation of the trained SVM to the new datasets. On the other hand, using non-constraint approach for initial training does not only improve the classification of the target set, but also preserve the accuracy of the initial dataset (see Table 33). This is important if we don't want to lose the classification accuracy of the initial training set such as the case when re-training the SVM to accommodate day and night time images. In both cases, the achieved improvement requires only several hundreds of additional datasets, which can be readily obtained. The cost involved and the resources required will therefore be minimal.

The use of incremental SVM here is critical. With incremental SVM, additional training vectors can be added to the learning process without retraining from scratch. Given that training is the most computationally intensive task in the classification problem, incremental SVM can provide a significant saving in training time. It also enables us to evaluate the contribution of the newly added vectors to the classification performance of the evolving SVM. As an application, we were able to constraint the support vectors that can be included into the solution set in terms of their contribution to the classification accuracy. This in turn allowed us to select only subsets within the initial training dataset that are useful for the optimization of the target set. Moreover,

the training process itself involves the selection process. And with an additional incremental re-training, the target dataset can be easily incorporated into the final SVM.

In conclusion, we have demonstrated the combined use of non-constraint/constraint initial training and incremental re-training to adapt SVM image classifiers to changes in imaging conditions. When applied to the car detection problem, significant improvement in the classification accuracy was achieved validating the efficacy of the method. The small number of additional datasets required for re-training makes the approach cost effective and practical for use in large deployment, such as in intelligent traffic systems, as the additional cost and needed resources are minimal.

References

- [1] Thrun, S., and Mitchell, T.M.: Learning one more thing. Proceedings of the 14th International Joint Conference on Artificial Intelligence (1995).
- [2] Caruana, R.: Multitask learning. *Machine Learning* 28(1), 41–75 (1997)
- [3] Dai, W., Yang, Q., Xue, G-R., and Yu, Y.: Boosting for Transfer Learning. Proceedings of the 24th International Conference on Machine Learning (2007)
- [4] Wu, P., and Dietterich, T.: Improving SVM Accuracy by Training on Auxiliary Data Sources. Proceedings of the 21st International Conference on Machine Learning (2004)
- [5] Raina, R., Battle, A., Lee, H., Packer, B., and Ng, A.: Self-taught Learning: Transfer Learning from Unlabeled Data. Proceedings of the 24th International Conference on Machine Learning (2007)
- [6] Cauwenberghs, G. and Poggio, T.: Incremental and Decremental Support Vector Machine Learning. In: Leen, T.K., Dietterich, T.G., and Tresp, V. (eds) *Advances in Neural Information Processing Systems*, vol. 13, pp. 409-415. MIT Press (2001)
- [7] Laskov, P., Gehl, C., Kruger, S., and Muller, K.R.: Incremental Support Vector Learning: Analysis, Implementation and Application. *Journal of Machine Learning Research* 7, 1909-1936 (2006)
- [8] Dalal, N. and Triggs, B.: Histograms of Oriented Gradients for Human Detection. *Conference on Computer Vision and Pattern Recognition* (2005)
- [9] Ralaivola, L. and d Alche Buc, F.: Incremental support vector machine learning: A local approach. *LNCS* 2130, 322-329 (2001)
- [10] Kivinen, J., Smola, A. J. and Williamson, R. C.: Online learning with kernels. In: Diettrich, T.G, Becker, S., and Ghahramani, Z. (eds.) *Advances In Neural Information Processing Systems (NIPS01)*. pp. 785-792 (2001)