Greedy-Based Design of Sparse Two-Stage SVMs for Fast Classification

Rezaul Karim, Martin Bergtholdt, Jörg Kappes, and Christoph Schnörr

University of Mannheim
Dept. Math & CS – CVGPR Group
karim@rumms.uni-mannheim.de

Abstract. Cascades of classifiers constitute an important architecture for fast object detection. While boosting of simple (weak) classifiers provides an established framework, the design of similar architectures with more powerful (strong) classifiers has become the subject of current research. In this paper, we focus on greedy strategies recently proposed in the literature that allow to learn sparse Support Vector Machines (SVMs) without the need to train full SVMs beforehand. We show (i) that asymmetric data sets that are typical for object detection scenarios can be successfully handled, and (ii) that the complementary training of two sparse SVMs leads to sequential two-stage classifiers that slightly outperform a full SVM, but only need about 10% kernel evaluations for classifying a pattern.

1 Introduction

Cascades of classifiers constitute an important architecture for fast object detection. A well-known and promiment example is the work of Viola and Jones [4] on face detection based on a cascade of boosted weak classifiers that only require simple image convolutions for feature extraction and thresholding. This framework is not directly applicable to kernel classifiers like support vector machines (SVMs), for instance, because boosting based on such strong classifiers as components is less effective. In many applications, however, the flexibility of kernel machines is a decisive advantage, as they can be applied to arbitrary features and pattern representations including histograms, sets, graphs, etc. This raises the question of how to design structured architectures for efficient classification using kernel machines as components.

Accordingly, this problem has spurred research recently. Related work can be roughly, but not disjointly, classified

- into approaches [6,5,8,11,1] to the design of *Reduced Support Vector Machines (RSVMs)* that require less computational costs than the standard SVM for classifying a pattern, and
- into approaches [10,5,7,9] that exploit SVMs (either reduced or not) as components of a structured architecture for classification.

Regarding the former class of approaches, RSVMs require only a *fraction of kernel evaluations* for classifying a pattern, either by computing a sparse subset

F.A. Hamprecht, C. Schnörr, and B. Jähne (Eds.): DAGM 2007, LNCS 4713, pp. 395–404, 2007.

of the support vectors of the full SVM [6,1], or by computing a novel small set of vectors in order to replace the support vectors altogether [5,11]. Additionally, wavelet approximations of these latter vectors have been investigated in [8] in order to *efficiently evaluate the arguments* (i.e. dot products between pattern vectors) to which the kernel function is applied.

The latter class of approaches, on the other hand, is focusing on *structured SVM-based classification* for face detection. Heisele et al. [10] studied a hierarchy of linear SVMs including a single nonlinear SVM as top node. Thresholds were tuned for optimizing classification performance and speed, followed by feature selection. Romdhani et al. [5] proposed a single chain of SVMs that is optimized also by threshold tuning, and by approximating a fully nonlinear SVM that has to be computed beforehand, whereas a decision tree with linear SVMs is suggested in [9]. Finally, Sahbi and Geman [7] recently presented a tree-structured hierarchy of SVMs that again is optimized by the reduced set technique used in [5] and threshold selection, and is operating on an application specific partitioning of the space of patterns (faces) according to different poses.

Contribution. In this paper, we assess two different direct greedy strategies [6,1] for designing reduced SVMs (RSVMs) in connection with the sequential combination of *two nonlinear RSVMs*. Such two-stage classifiers form the core of any recursively designed larger structured architecture. Figure 1 illustrates the basic idea underlying the design of RSVMs.

The rationale behind our choice is as follows: Firstly, we focus on *direct* RSVM computation rather than on approximations of fully nonlinear SVMs, in order to avoid the need to train the latter beforehand. Secondly, we refrain from the computation of novel representatives of support vectors as done in [5,11] because this relies on complex optimization problems that are sensitive to initialization, step sizes, etc. Corresponding problems can easily interfere with our main objective, the assessment of structured architectures to classification using RSVMs. Finally, in order to meet error rate specifications, we prefer training with asymmetric costs over threshold tuning because the latter is known to result in classifiers that are not ROC-optimal [3].

Organization. The two greedy strategies [6,1] for designing RSVMs are described in sections 2 and 3. We slightly modified the latter approach by including a bias term (threshold) into the RSVM decision function that is also determined during training. In section 4, we report the results of numerical experiments addressing the following aspects: Validation of the implementation using standard benchmark data, performance evaluation for fixed classifier complexities, coping with asymmetric data and training costs, complementary design of two-stage RSVM classifier.

Notation. False/true and negative/positive error rates are abbreviated with FNR, FPR, TNR and TPR, respectively, and expressed as percentage %. $k(x, y)$ denotes an admissible kernel function (e.g. Gaussian), K_m a $m \times m$ kernel matrix, and $k_m(x)$ the vector $k_m(x) = (k(x_1, x), \ldots, k(x_m, x))^\top$.

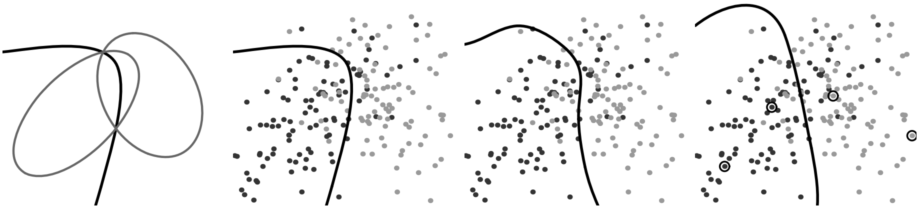

Fig. 1. Illustration of RSVMs. From left to right: **(i)** Level-lines of two Gaussian distributions and the decision line of the Bayesian classifier. The Bayes error is $L^* \approx 16.465\%$. **(ii)** The Bayesian classifier and a sample of 200 patterns. **(iii)** The decision surface of the standard SVM trained with all sample patterns, and with optimized parameters. The number of support vectors is 93. The error rate is $L_{full} \approx 17.005\%$. **(iv)** A RSVM with 4 support vectors indicated by circles. The error rate $L_{red} \approx 17.71\%$ is only slightly worse than that of the standard SVM, whereas the kernel evaluations have been reduced by a factor of about 23.

2 RSVM-1: Design by Feature Subset Search

The approach of Franc and Hlavac [6] to the design of RSVMs with fixed classifier complexity involves two phases that we describe next:

(i) Search for an optimal subset of samples $\mathcal{X}_m \subset \mathcal{X}_n = \{x_1, \dots, x_n\}\,,\ m \ll n$.
(ii) Compute a classifier with computational costs proportional to the evaluation of $k_m(x)$.

Let $\phi(\cdot)$ denote the feature mapping induced by $k(x,y)$, that is $k(x,y) = \phi(x) \cdot \phi(y)$. To simplify notation, we treat $\phi(\cdot)$ as any other vector.

In phase (i), the subset $\mathcal{X}_m$ is iteratively determined as $\mathcal{X}_r = \{x_1, \dots, x_r\} \subset \mathcal{X}_n$ for $r = 2, \dots, m$, $m < n$, such that for each $r < m$, the next pattern to be included satisfies

$$x_{r+1} = \operatorname*{argmax}_{x \in \mathcal{X} \setminus \mathcal{X}_r} d_r(x)\ ,$$

where $d_r(x)$ is the distance of $\phi(x)$ to the subspace spanned by $\phi(x_1), \dots, \phi(x_r)$. It is straightforward to check that this distance between $\phi(x)$ and its orthogonal projection $P_r \phi(x)$ is given by

$$\begin{aligned} d_r^2(x) &= \|\phi(x) - P_r \phi(x)\|^2 \\ &= k(x,x) - 2 k_r(x)^\top \beta_x + \beta_x^\top K_r \beta_x\ , \qquad \beta_x = K_r^{-1} k_r(x)\ . \end{aligned}$$

After termination of the greedy search $\mathcal{X}_m$ is given, and based on the Cholesky factorization $K_m = U^\top U$, all feature vectors $\phi(x_i)\,,\ x_i \in \mathcal{X}_n$, are approximated by their projections

$$\big(P_m \phi(x_i)\big) \cdot \big(P_m \phi(x_j)\big) = \beta_i^\top K_m \beta_j = \beta_i^\top U^\top U \beta_j =: \gamma_i^\top \gamma_j\ .$$

As a result, each training pattern $x_i \in \mathcal{X}_n$ is represented by a vector $\gamma_i \in \mathbb{R}^m$. In phase (ii) of the approach, we compute a standard SVM with $\mathcal{X}$ replaced by $\Gamma := (\gamma_1, \ldots, \gamma_n)$:

$$\min_{w,b} \left\{ \frac{1}{2}\|w\|^2 + C_+ \sum_{y_i>0} \xi_i + C_- \sum_{y_i<0} \xi_i \right\} , \quad \text{s.t.} \quad D_y(\Gamma^\top w + be) \geq e - \xi \ , \ \xi \geq 0 \ .$$

Here, D_y denotes the diagonal matrix with the class variables $y_i \in \{+1, -1\}$. In order to classify a novel pattern x, we compute its representative $\gamma_x = U\beta_x = UK_m^{-1}k_m(x)$ and evaluate the decision function

$$f_m(x) = w^\top \gamma_x + b \ , \quad w = \Gamma D_y \alpha = \sum_{i=1}^{n_s} \alpha_i y_i \gamma_i \ .$$

Re-inserting the definitions of γ_x and γ_i, these two steps amount to compute

$$f_m(x) = \sum_{i=1}^{n_s} \alpha_i y_i \beta_i^\top k_m(x) + b \ ,$$

with n_s denoting the number of support vectors. Note that the computational complexity is dominated by the *fixed* number of m kernel evaluations $k_m(x)$.

3 RSVM-2: Direct Greedy-Based Design

We outline a slight modification of the approach [1]. The modification concerns asymmetric training costs and the inclusion of a bias b into the decision function

$$f_m(x) = w^\top \phi(x) + b = \beta^\top k_m(x) + b \ , \quad w = \sum_{i=1}^{m} \beta_i \phi(x_i) \ .$$

Similar to the previous section, the basic idea is to perform a greedy search of an optimal subset $\phi(x_1), \ldots, \phi(x_m)$ in feature space, and to train directly a RSVM by minimizing the primal objective function

$$\begin{aligned} E(\beta, b) = \frac{1}{2}\beta^\top K_m \beta + \frac{C_+}{2} \sum_{y_i>0} \max \big\{0, 1 - \big(\beta^\top k_m(x) + b\big)\big\}^2 \\ + \frac{C_-}{2} \sum_{y_i<0} \max \big\{0, 1 + \big(\beta^\top k_m(x) + b\big)\big\}^2 \end{aligned}$$

with the following Newton-like iteration: Let k be the iteration counter, (β^k, b^k) the current iterate, and I_+ and I_- denote the indices of training patterns whose regularization term does not vanish: $1 - y_i\big(\beta^k \cdot k_m(x) + b^k\big) > 0$. Then we compute

$$(\beta^{k+1/2}, b^{k+1/2}) = (\beta^k, b^k) - \big[H_E(\beta^k, b^k)\big]^{-1} \nabla E(\beta^k, b^k)$$

followed by the line search

$$(\beta^{k+1}, b^{k+1}) = \operatorname*{argmin}_{t \in [0,1]} E\Big((1-t)(\beta^k, b^k) + t(\beta^{k+1/2}, b^{k+1/2})\Big) .$$

The gradient and the Hessian are given by

$$\nabla E(\beta, b) = \begin{pmatrix} K_m \beta + C_+ K_{I_+,m}^\top (f_{I_+} - y_{I_+}) + C_- K_{I_-,m}^\top (f_{I_-} - y_{I_-}) \\ C_+ e_{I_+}^\top (f_{I_+} - y_{I_+}) + C_- e_{I_-}^\top (f_{I_-} - y_{I_-}) \end{pmatrix} ,$$

$$H_E(\beta, b) = \begin{pmatrix} K_m + C_+ K_{I_+,m}^\top K_{I_+,m} + C_- K_{I_-,m}^\top K_{I_-,m} & C_+ K_{I_+,m}^\top e_{I_+} + C_- K_{I_-,m}^\top e_{I_-} \\ C_+ e_{I_+}^\top K_{I_+,m} + C_- e_{I_-}^\top K_{I_-,m} & C_+ |I_+| + C_- |I_-| \end{pmatrix} ,$$

where $|I_+|, |I_-|$ denote the respective number of indices, e_{I_+}, e_{I_-} are one-vectors of the corresponding dimensions, and f_{I_+}, f_{I_-} and $K_{I_+,m}, K_{I_-,m}$ are vectors and matrices, respectively, formed by selecting decisions function values $f_m(x_i)$ and the rows of K_m as indexed by I_+, I_-.

During the greedy search procedure, this parameter fitting is done for increasing dimensions $r = 2, \ldots, m$. For fixed r and minimizing parameters (β, b) , $\beta \in \mathbb{R}^r$, the criterion for selecting the next pattern x_{r+1} is the largest change of the energy determined by optimizing the two variables $\min_{\beta_{r+1}, b} E(\beta_{r+1}, b)$ with $\beta = (\beta_1, \ldots, \beta_r)$ kept fixed.

4 Numerical Performance Evaluation

This section summarizes our experimental evaluation of RSVM-1 and RSVM-2 under various aspects. With the exception of the real data experiment reported in subsection 4.5, all experiments were conducted with computer-generate data in order to determine the test error rates accurately.

4.1 Validation of the Implementation

The approach described in section 3 reproduced the performance measures reported in [1] for the benchmark data sets [2]. For example, Figure 2 shows the average error rate (%) as a function of the number of support vectors.

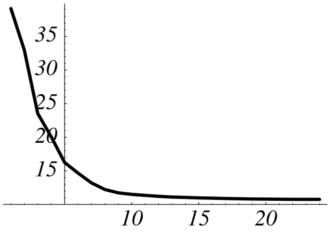

Fig. 2. Average error rate (%) of the RSVM-2 for the Banana data set [2] as a function of the number m of support vectors

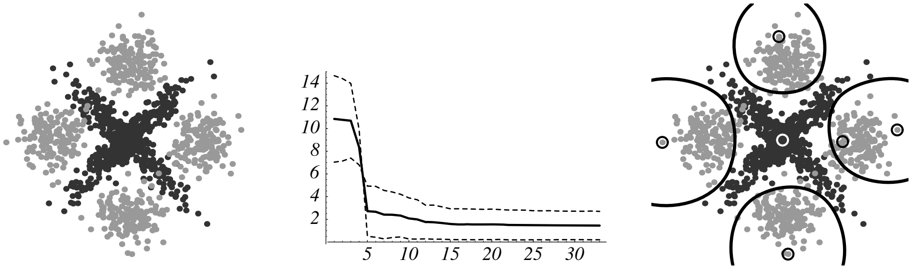

Fig. 3. Left: Training set for a 2-class problem. The standard SVM returns FNR/FPR(%)=0.45/2.12 and 68 support vectors. **Center:** Error rate (thick line) and FNR/FPR (thin lines) for the RSVM-2 that significantly outperforms RSVM-1 (cf. Table 1). **Right:** The first 6 SVs and the corresponding decision line of the RSVM-2 (FNR/FPR(%)=0.51/4.93). On the average, RSVM-2 shows an acceptable performance at about 10% computational costs of a standard SVM.

Table 1. Performance of the reduced SVMs for various fixed classifier complexities. RSVM-2 considerably outperforms RSVM-1. The standard SVM returns FNR/FPR(%)=0.45/2.12 and 68 support vectors.

# SVs	4	10	15	20	30
RSVM-1: FNR/FPR (%)	47.86/56.13	40.74/18.01	14.44/13.93	2.14/8.48	0.39/2.50
RSVM-2: FNR/FPR (%)	13.98/7.40	0.45/4.21	0.25/3.10	0.22/2.88	0.21/2.72

4.2 Performance for Fixed Classifier Complexity

In compare the different greedy strategies underlying RSVM-1 and RSVM-2, respectively, we fixed the classifier complexities to $m \in \{4, 10, 15, 20, 30\}$ and evaluated the FNR and FPR of both reduced machines. The details are given in Figure 3 and Table 1.

It turned out that RSVM-2 is consistently superior to RSVM-1 and shows an performance comparable to the full SVM while needing only 10% of the number of support vectors on the average.

4.3 Asymmetric Training Data

We performed an evaluation of RSVM-1 and RSVM-2 similar to the previous section, but with asymmetric training sets and asymmetric training costs. This situation is typical for detection scenarios where a large number of background patterns are easily available for training, whereas the number of object patterns is limited. A priori, it is not clear whether greedy search breaks down in such situations. Figure 4 and Table 2 provide the quantitative details.

While the RSVM-2 perform as well as in the symmetric case (previous subsection), the performance of the RSVM-1 becomes even worse. Likewise, the

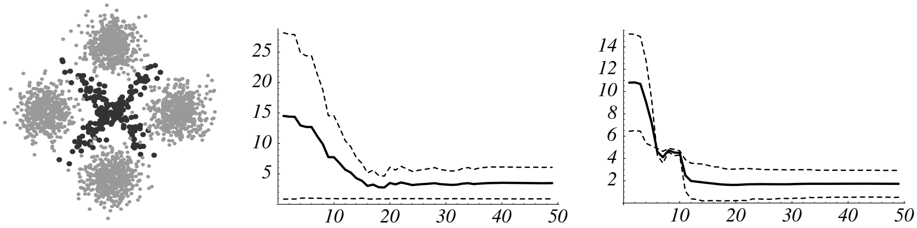

Fig. 4. Left: An *asymmetric* training set (fore-/background samples = 1/10). The standard SVM returns FNR/FPR(%)=3.79/0.86 and 92 support vectors for symmetric training costs, and FNR/FPR(%)=1.08/2.16 and 153 support vectors for an asymmetric choice of the training costs. **Center:** Error rate (thick line) and FNR/FPR (thin lines) for the RSVM-2 trained with symmetric costs, that significantly ourperforms RSVM-1 also in such asymmetric scenarios (cf. Table 2). The greedy optimization, however, mainly focuses on the larger background sample set (lower dashed line), yielding a suboptimal overall performance **Right:** Asymmetric training costs enables to steer the greedy search and to optimize the overall performance (note that the ordinate-scale differs from the figure in the middle). For 15 support vectors, that is about 10% classification costs of the full SVM, the RSVM-2 returns FNR/FPR(%)=0.2/3.5.

Table 2. Performance of the reduced SVMs for various fixed classifier complexities and *asymmetric* training costs. RSVM-2 considerably outperforms RSVM-1. The standard SVM returns FNR/FPR(%)=1.08/2.16 and 153 support vectors.

# SVs	4	10	15	20	30
RSVM-1: FNR/FPR (%)	54.59/32.15	51.74/12.40	43.00/10.79	1.87/6.34	0.87/2.85
RSVM-2: FNR/FPR (%)	14.92/6.44	4.25/4.82	0.2/3.5	0.21/3.03	0.41/2.98

relationship of approximation quality and computational complexity of the RSVM-2 relativ to the fully nonlinear SVM did not change noticeably.

4.4 Two-Stages Sparse SVM Classification

The objective of this section is to show that in principle two sparse SVMs can be combined sequentially without loss of classification performance, but at considerably reduced computational classification costs. Being inferior to RSVM-2, we did not consider RSVM-1 in this context, and we simply denote RSVM-2 by RSVM in this section. We use subscripts 1 and 2 for the RSVM at stage 1 and 2, respectively.

Figure 5, left and right, show two RSVMs designed as stage-1 and stage-2 classifiers. The RSVM at stage 1 was asymmetrically designed so as to yield a very low FNR. For the stage-2 RSVM, only those background patterns were used for training that were accepted as false positives at stage 1. This is reasonable

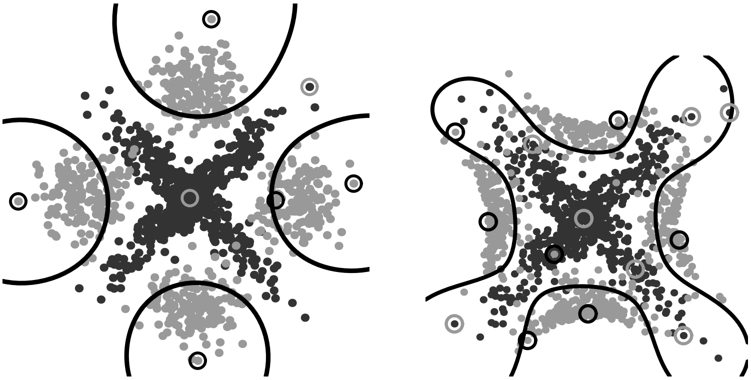

Fig. 5. Sequential classification by two RSVM-2. Both stages were trained by direct greedy optimization. **Left:** Asymmetrically trained stage-1 RSVM with minimal FNR (fraction of missed objects). **Right:** Stage-2 RSVM asymmetrically trained on positive examples and false positives accepted by the stage-1 RSVM. The overall performance is FNR/FPR(%) = 0.52/0.45 (see text for more details).

because in practice typically a large number of background patterns are available. The performance data of the two classifiers are:

$$\mathrm{FNR}_1/\mathrm{FPR}_1(\%) = 0.0035/15.83 \qquad (\#\mathrm{SV} = 7)$$
$$\mathrm{FNR}_2/\mathrm{FPR}_2(\%) = 0.51/2.84 \qquad (\#\mathrm{SV} = 14)$$

Then the overall performance is

$$\mathrm{FNR} = \mathrm{FNR}_2\mathrm{TPR}_1 + \mathrm{FNR}_1 \approx 0.51\% \,, \qquad \mathrm{FPR} = \mathrm{FPR}_2\mathrm{FPR}_1 \approx 0.45\% \,,$$

which compares favourably with the full SVM (see the caption of Figure 3).

The average computational costs per pattern are largely dominated by the first-stage classifier which typically requires 10% computation time relative to the full SVM. Assuming that the second RSVM has twice the number of support vectors, that is 20%, and that an object occurs at 0.1% of all image locations, than the two-stage classifier requires on the average about

$$0.2\,(0.001\,\mathrm{TPR}_1 + 0.999\,\mathrm{FPR}_1) + 0.1 \approx 13\%$$

of the computation time of the full SVM.

Evaluation of benchmark data. We also applied the two-stages classifier combining a very sparse RSVM at stage 1, followed by the RSVM designed as reported in section 4.1, using the bechmark data[2], and averaged the results over the corresponding 100 training-test pairs of data sets. The effective number of support vectors is the sum of #SVs of the first machine plus #SVs of the second machine multiplied by the acceptance rate of the first machine.

Table 3 shows that in comparison to [1] the classification cost can be further reduced without a significance loss of performance.

Table 3. Benchmark evaluation of the two-stages sparse SVM. The effective number of SVs minimizes the classification costs and yields comparable classification performance.

Dataset	Our Cascade		Keerthi *et al* [1]		SVM[1]	
	Effictive #SVs	Error	#SVs	Error	#SVs	Error
Breast	4.67(0.66)	26.80(4.92)	12.1(5.6)	29.22(2.11)	185.8(16.44)	28.18(3.00)
Diabetis	6.48(0.54)	26.32(2.28)	13.8(5.6)	23.47(1.36)	426.3(26.91)	23.73(1.24)
German	4.41(0.63)	27.80(2.45)	14.0(7.3)	24.90(1.50)	630.4(22.48)	24.47(1.97)
Ringnorm	9.79(0.19)	2.04(0.28)	12.9(2.0)	1.97(0.57)	334.9(108.54)	1.68(0.24)
Thyroid	5.45(0.41)	5.61(2.41)	10.6(2.3)	5.47(0.78)	57.80(39.61)	4.93(2.18)
Waveform	9.16(0.40)	12.75(1.33)	14.4(3.3)	10.66(0.99)	246.9(57.80)	10.04(0.67)

4.5 Real Data

Although specific applications are not within the scope of this paper, we report the performance of the RSVM-2 for an experiment with real data, to assure that the findings reported above generalize to other data sets.

For a real-world challenge, we considered head detection on a set of 1042 images containing humans in various poses at approximately the same scale. We divided the data-set into 603 training images and 439 test images, such that there are no two images of individuals under similar conditions in the test set, and all are mutually distinct to the training set. As a result, we may expect realistic general performance measures. From these images, small patches of size 32x32 were extracted at the head location (provided by the user), and the popular SIFT-features [12] were computed from the patches. We used 4x4 location and 8 orientation bins resulting in 128-dimensional feature vectors. Contrary to the original formulation, we did no local orientation or scale normalization.

For the background, we computed 9934 features at locations not containing any heads, which we divided into 4967 training and 4967 test features. Note the asymmetry in the data, with a ratio of background/foreground of $\approx 8/1$ for training and $\approx 11/1$ for testing.

Training a fully nonlinear SVM with asymmetric costs resulted in 1720 support vectors and error rates FNR/FPR (%) = 14.35/1.39. The RSVM-2 showed a comparable performance, FNR/FPR (%) = 10.02/4.11, for only 47 support vectors, however, that is with classification costs reduced by a factor of about 36!

5 Conclusions

We compared two greedy strategies recently proposed for the direct design of reduced nonlinear SVMs. One of these strategies, suggested in [1], performed uniformly well irrespective of the nature of the data set, and also in asymmetric situations that are typical for object detection scenarios.

It should be pointed out that the factor of decreasing computational costs reported in this paper has to be *multiplied* by the acceleration factors reported in [8], that are obtained by an independent technique as discussed in section 1.

We showed that the complementary design of two reduced SVMs results in sequential two-stage classifiers that may even outperform fully nonlinear SVMs. Such classifiers may form the core of larger structured classifiers using RSVMs as components. This will be investigated in our future work.

Acknowledgement. This work has been supported by the European Marie Curie Research Training Network VISIONTRAIN (contract number MRTN-CT-2004-005439).

References

1. Keerthi, S.S., Chapelle, O., DeCoste, D.: Building support vector machines with reduced classifier complexity. J. Mach. Learning Res. 7, 1493–1515 (2006)
2. Rätsch, G.: Benchmark data sets, `http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm`
3. Bach, F.R., Heckerman, D., Horvitz, E.: Considering cost asymmetry in learning classifiers. J. Mach. Learning Res. 7, 1713–1741 (2006)
4. Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comp. Vision 57(2), 137–154 (2004)
5. Romdhani, S., Torr, P., Schölkopf, B., Blake, A.: Efficient face detection by a cascaded support-vector machine expansion. Proc. Royal Soc. A 460, 3283–3297 (2004)
6. Franc, V., Hlavać, V.: Greedy algorithm for a training set reduction in the kernel methods. In: Petkov, N., Westenberg, M.A. (eds.) CAIP 2003. LNCS, vol. 2756, pp. 426–433. Springer, Heidelberg (2003)
7. Sahbi, H., Geman, D.: A hierarchy of support vector machines for pattern detection. J. Mach. Learning Res. 7, 2087–2123 (2006)
8. Rätsch, M., Romdhani, S., Teschke, G., Vetter, T.: Over-complete wavelet approximation of a support vector machine for efficient classification. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) Pattern Recognition. LNCS, vol. 3663, pp. 351–360. Springer, Heidelberg (2005)
9. Zapién, K., Fehr, J., Burkhardt, H.: Fast support vector machine classification using linear SVMs. In: 18th International Conference on Pattern Recognition (ICPR 2006), vol. 3, pp. 366–369. IEEE, Los Alamitos (2006)
10. Heisele, B., Serre, T., Prentice, S., Poggio, T.: Hierarchical classification and feature reduction for fast face detection with support vector machines. Pattern Recognition 36(9), 2007–2017 (2003)
11. Wu, M., Schölkopf, B., Bakir, G.: A direct method for building sparse kernel learning algorithms. J. Mach. Learning Res. 7, 603–624 (2006)
12. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comp. Vision 60(2), 91–110 (2004)