Minimal Correlation Classification

Noga Levy and Lior Wolf

The Blavatnik School of Computer Science, Tel Aviv University, Israel

Abstract. When the description of the visual data is rich and consists of many features, a classification based on a single model can often be enhanced using an ensemble of models. We suggest a new ensemble learning method that encourages the base classifiers to learn different aspects of the data. Initially, a binary classification algorithm such as Support Vector Machine is applied and its confidence values on the training set are considered. Following the idea that ensemble methods work best when the classification errors of the base classifiers are not related, we serially learn additional classifiers whose output confidences on the training examples are minimally correlated. Finally, these uncorrelated classifiers are assembled using the GentleBoost algorithm. Presented experiments in various visual recognition domains demonstrate the effectiveness of the method.

1 Introduction

In classification problems, one aims to learn a classifier that generalizes well on future data from a limited number of training examples. Given a labeled training set $(x_i, y_i)_{i=1}^m$, a classification algorithm learns a predictor from a predefined family of hypotheses that accurately labels new examples. An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way, typically by a weighted voting, to classify the new examples. Effective ensemble learning obtains better predictive performance than any of the individual classifiers [14], but this is not always the case [7].

As described in [6], a necessary and sufficient condition for an ensemble of classifiers to be more accurate than any of its individual members is that the classifiers are accurate and diverse. Classifiers are diverse if their classification errors on the same data differ. As long as each classifier is reasonably accurate, the diversity promotes the error correcting ability of the majority vote.

Liu and Yao proposed Negative Correlation Learning (NCL) in which individual neural networks are trained simultaneously [18]. They added a penalty term to produce learners whose errors tend to be negatively correlated. The cost function used to train the l^{th} neural network is $e_l = \sum_{k=1}^m \left(f_l(x_k) - y_k\right)^2 - \lambda \sum_{k=1}^m \left(f_l(x_k) - f_{ensemble}(x_k)\right)^2$, where f_l is the classification function of the l^{th} neural network, and $f_{ensemble}$ is the combined one. That is, NCL encourages each learner to differ from the combined vote on every training sample. The NCL principle is adapted to various setting, for example, Hu and Mao designed an NCL-based ensemble of SVMs for regression problems [12].

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part VI, LNCS 7577, pp. 29–42, 2012.

MMDA is a dimensionality reduction algorithm that also relies on the minimal correlation principle [15]. SVM is repeatedly solved such that at each iteration the separating hyperplane is constrained to be orthogonal to the previous ones. In our tests it became apparent that such an approach is too constraining and does not take into account the need to treat each class separately.

Our approach measures correlation between every pair of classifiers per each class of the dataset separately and employs a different decorrelation criterion. The minimal correlation method suggested here learns the base classifiers successively using the same training set and classification algorithm, but every classifier is required to output predictions that are uncorrelated with the predictions of the former classifiers. The underlying assumption is that demanding the predictions of one classifier to be uncorrelated with the predictions of another (in addition to the accuracy demand) leads to distinct yet complementing models,+ even though the classifiers are not conditionally independent.

2 Minimal Correlation Ensemble

The training data $S = \{(x_i, y_i)\}_{i=1}^m$ consists of examples $x_i \in \mathbb{R}^m$ and binary labels $y_i \in \pm 1$. Our first classifier, denoted w_1, is the solution of a regular SVM optimization problem:

$$\begin{aligned} &\min_w \ \tfrac{\lambda}{2}\|w\|^2 + \textstyle\sum_{i=1}^m \xi_i \\ &\text{s.t.} \\ &\qquad \forall i.\ \ y_i(w^T x_i) \geq 1 - \xi_i \ \ , \ \ \xi_i \geq 0. \end{aligned}$$

Let X_p be the set of all positive examples and X_n the set of all negative examples in S, arranged as matrices. We now look for a second classifier, w_2, with a similar objective function, but add the correlation demand: the predictions of w_2 on the examples of each class (separately) should be uncorrelated with the predictions of w_1 on these examples. The separation to classes is necessary since all accurate classifiers are expected to be correlated as they provide comparable labeling. Employing Pearson's sample correlation, this demand translates into minimizing

$$\begin{aligned} r_p &= \frac{\sum_{i \in I_p} \left(w_1^T x_i - \overline{w_1^T X_p}\right)\left(w_2^T x_i - \overline{w_2^T X_p}\right)}{(n-1)s_1 s_2} = \\ &= \frac{\langle w_1^T \left(X_p - \overline{X_p}\right), w_2^T \left(X_p - \overline{X_p}\right)\rangle}{\|w_1^T \left(X_p - \overline{X_p}\right)\| \|w_2^T \left(X_p - \overline{X_p}\right)\|} \end{aligned}$$

where I_p is the set of indices of the positive examples, $\overline{X_p}$ is a matrix whose columns are the mean vector of the positive class, $\overline{w_i^T X_p}$ and s_i for $i = \{1, 2\}$ are the mean and standard deviation of $w_i^T X_p$, respectively.

Let $\hat{y}_{p_i}$ be the normalized predictions of w_i on X_p, that is $\hat{y}_{p_i} = \frac{w_i^T \left(X_p - \overline{X_p}\right)^T}{\|w_i^T \left(X_p - \overline{X_p}\right)\|}$. To maintain convexity, s_2 is omitted and the additional expression becomes the covariance between the normalized output of the existing classifiers and the output of the new classifier,

$$r_p = \langle \hat{y}_{p_1}, w_2^T \left(X_p - \overline{X_p}\right)\rangle \ .$$

Denote $v_p = \hat{y}_{p_1}\left(X_p^T - \overline{X_p}\right)$ and $v_n = \hat{y}_{n_1}\left(X_n^T - \overline{X_n}\right)$. Since we are interested in the correlation with the minimal magnitude, we choose to add the terms $r_p^2 + r_n^2 = \|w^T v_p\|^2 + \|w^T v_n\|^2$ to the objective function.
The tradeoff between the correlation and the other expressions in the objective functions is controlled by a tradeoff parameter η. The new optimization problem whose solution results in a second classifier is

$$\begin{aligned} &\min_w \tfrac{\lambda}{2}\|w\|^2 + \tfrac{\eta}{2}\left(\|w^T v_p\|^2 + \|w^T v_n\|^2\right) + \textstyle\sum_{i=1}^m \xi_i \\ &\text{s.t.} \quad \forall i.\ y_i\langle w, x_i\rangle \geq 1 - \xi_i \ \ , \ \ \xi_i \geq 0. \end{aligned}$$

The optimization problem constructed to learn the k^{th} classifier contains $2(k-1)$ correlation expressions, one per each preceding classifier and per each class,

$$\begin{aligned} &\min_w \frac{\lambda}{2}\|w\|^2 + \sum_{j=1}^{k-1} \frac{\eta_j}{2}\left(\|w^T v_{p_j}\|^2 + \|w^T v_{n_j}\|^2\right) + \sum_{i=1}^{m} \xi_i \\ &\text{s.t.} \qquad \forall i.\ y_i\langle w, x_i\rangle \geq 1 - \xi_i \ \ , \ \ \xi_i \geq 0. \end{aligned} \tag{1}$$

The computational cost of solving all optimization problems is linear in the number of classifiers.

2.1 The Dual Problem

We derive the dual problem to get a kernelized version of our method. Let α_i be the dual variable of the margin constraint of example (x_i, y_i) and β_i the dual variable of the non-negativity constraint of ξ_i. The Lagrangian of optimization function (Eq. 1) is

$$\begin{aligned} \mathcal{L}(w, \alpha, \beta) = &\frac{\lambda}{2}\|w\|^2 + \sum_{j=1}^{k-1} \eta_j \left(\|w^T v_{p_j}\|^2 + \|w^T v_{n_j}\|^2\right) \\ &+ \sum_{i=1}^{m} \xi_i + \sum_{i=1}^{m} \alpha_i(1 - \xi_i - y_i w^T x_i) - \sum_{i=1}^{m} \beta_i \xi_i. \end{aligned}$$

The w that minimizes the primal problem can be expressed as a function of the dual variables α and β by solving $\partial\mathcal{L}/\partial w = 0$,

$$w = \left(\lambda I + \sum_{j=1}^{k-1} \eta_j \left(v_{p_j} v_{p_j}^T + v_{n_j} v_{n_j}^T\right)\right)^{-1} \sum_{i=1}^{m} \alpha_i y_i x_i^T \ .$$

Denote by V the matrix whose columns are the vectors v_{p_i} and v_{n_i} multiplied by the square root of the matching tradeoff parameter η_i,

$$V = \left[\sqrt{\eta_1} v_{p_1}, \sqrt{\eta_1} v_{n_1}, \ldots, \sqrt{\eta_{k-1}} v_{p_{k-1}}, \sqrt{\eta_{k-1}} v_{n_{k-1}}\right], \tag{2}$$

then

$$w = \left(\lambda I + V V^T\right)^{-1} \sum_{i=1}^{m} \alpha_i y_i x_i \ . \tag{3}$$

Substituting the inverse matrix in Eq.3 by the Woodbury identity, and then using the fact that the columns of V are multiplications of training examples, classifying an example x can be kernelized using any kernel function $k(x_i, x_j)$ so that $w^T x$ is

$$\lambda \sum_{i=1}^{m} \alpha_i y_i [k(x_i, x) - \sqrt{\eta}^T k(x_i, V)(I + V^T V)^{-1} k(x, V)^T \sqrt{\eta}],$$

where $k(x, V)$ is the column vector whose i^{th} coordinate is $k(x, V_i)$.

As in SVM, w is a weighted combination of the training examples, but in our solution this combination is projected onto a linear space that is determined by the correlation expressions. Using Eq. 3, the dual objective function can be kernelized as well. Denote $u = \sum_{i=1}^{m} \alpha_i y_i k(V, x_i)\sqrt{\eta}$, then the objective function becomes

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - u^T (I + V^T V)^{-1} u \ .$$

This formulation implies that the minimal correlation classifiers can be trained on high dimensional data using a kernel function.

2.2 Online Version

Following Pegasos (Primal Estimated sub-Gradient SOlver for SVM) [21], we derive an online sub-gradient descent algorithm for solving the optimization problem at hand. In iteration t of the algorithm, Pegasos chooses a random training example (x_{i_t}, y_{i_t}) by picking an index $i_t \in 1, \ldots, m$ uniformly at random. The objective function of the SVM is replaced with an approximation based on the training example (x_{i_t}, y_{i_t}), yielding:

$$f(w, i_t) = \frac{\lambda}{2}\|w\|^2 + l(w, (x_{i_t}, y_{i_t})) \ ,$$

where $l(w, (x, y)) = max(0, 1 - y_i w^T x_i)$ is the hinge-loss function and the update rule is a gradient decent step on the approximated function:

$$w_{t+1} = \left(1 - \frac{1}{t}\right) w_t + \beta \mathbb{1}[y_{i_t} \langle w_t, x_{i_t} \rangle < 1] \ y_{i_t} x_{i_t}$$

We follow a similar path with our objective function, and differently from the batch mode where all the training set is available, we receive one example at a time. The hinge loss function suits this setting since it is the sum of m expressions, each depends on one example. The correlation, on the other hand, depends on all the examples combined, and therefore cannot be separated into single example loss functions. That is, optimizing $\langle w_1^T x_i, w_2^T x_i \rangle$ is meaningless and we need to calculate the outputs of all training examples (or at least a large enough subset) to estimate the correlation.

Nevertheless, we can serially calculate the correlation of the already seen examples in a time and space efficient manner as described bellow. Denote by $X_{p,t}$ the matrix of positive training examples and by $X_{n,t}$ the matrix of the negative ones chosen in the iterations 1..t (note that during the run of each iteration only one training example is held). We define correlation of w_1 and w over the $X_{p,t}$ as

$$r_{p,t} = \frac{\left\langle w^T \left(X_{p,t} - \overline{X_{p,t}}\right), w_1^T \left(X_{p,t} - \overline{X_{p,t}}\right)\right\rangle}{\|w_1^T \left(X_{p,t} - \overline{X_{p,t}}\right)\|} =$$

$$= w^T \frac{\left(X_{p,t}X_{p,t}^T w_1 - t^2 \overline{X_{p,t}}\ \overline{X_{p,t}}^T w_1\right)}{\|w_1^T \left(X_{p,t} - \overline{X_{p,t}}\right)\|}.$$

Denote $v_{p,t} = \left(X_{p,t} - \overline{X_{p,t}}\right) \frac{w_1^T \left(X_{p,t} - \overline{X_{p,t}}\right)}{\|w_1^T \left(X_{p,t} - \overline{X_{p,t}}\right)\|}$. By maintaining the mean vector $\overline{X_{p,t}}$, the sum of squares $\sum (w_1^T X_{p,t})^2$, and the product $X_{p,t}X_{p,t}^T w_1$, the calculation of $v_{p,t+1}$ is time efficient. Denote $v_{n,t}$ similarly for the negative examples. If $y_{i_t} = 1$ then $v_{p,t}$ is updated while $v_{n,t} = v_{n,t-1}$, otherwise $v_{n,t}$ is calculated and $v_{p,t} = v_{p,t-1}$. Our approximation in iteration t is

$$f(w, i_t) = \frac{\lambda}{2}\|w\|^2 + \frac{\eta}{2}\left(\|w^T v_{p,t}\|^2 + \|w^T v_{n,t}\|^2\right) + l(w; (x_{i_t}, y_{i_t}))\ .$$

and the update step is computed in $O(d)$ time as

$$w_{t+1} = (1 - \frac{1}{\lambda t})\left(\lambda + \eta(v_{p,t}v_{p,t}^T + v_{n,t}v_{n,t}^T)\right) w_t + \beta \mathbb{1}[y_{i_t}\langle w_t, x_{i_t}\rangle < 1]\ y_{i_t}x_{i_t}$$

After a predetermined number T of iterations, we output the last iterate w_{T+1} .

2.3 Ensemble Method

After learning multiple classifiers, the base classifiers are assembled into one strong classifier by using a form of stacking [23]. We found that the GentleBoost shows the best improvement in performance. Assume we have k classifiers, $w_1, \ldots, w_k$ and a validation set $S = (x_i, y_i)_{i=1}^m$, and we want to combine these k classifiers into one strong classifier. First, we classify the validation set S using the k base classifiers. Every sample in S can now be represented as a vector in $\mathbb{R}^k$ whose coordinates are the k outputs, $t_i = (w_1^T x_i, \ldots, w_k^T x_i)$.

We regard these vectors t_i as new k-dimensional feature vectors, and learn a new classifier based on '$S' = (t_i, y_i)_{i=1}^m$. As this second stage classifier we use GentleBoost [9] over decision stumps. GentleBoost is a version of AdaBoost that uses a more conservative update of the weights assigned to the weak classifiers.

The classification of an unknown example x is as follows: we first classify x with $w_1, \ldots, w_k$ to construct the GentleBoost input vector t and then apply the ensemble classifier on t.

3 Generalization Bounds

Since every classifier is constrained by the preceding ones, a natural concern is that the performance of subsequent classifiers could diminish. We prove that the performance of the added classifiers is comparable to preceding ones. The optimization problem solved by the minimal correlation ensemble, after the first round, minimizes the norm of w as well as the norm the correlation expressions that depend on the training data and the preceding classifiers. Using the matrix V defined in Eq. 2, whose columns are the vectors multiplying w in all the correlation expressions that appear in the objective function, this norm is $\|w^T V\|$. A column V_i in V, is a multiplication of $\hat{y}_{\{p/n\}_j}$ (the normalized predictions of classifier j on the training examples in one of the classes) by $\sqrt{\eta_j} X_{\{p/n\}}$. The dependency on the former classifiers is derived from $\hat{y}_{\{p/n\}_j}$ and the dependency on the training data is derived from both $\hat{y}_{\{p/n\}_j}$ and $X_{\{p/n\}}$.

Deriving a generalization bound using Rademacher complexity is not applicable since the *iid* assumption, required to bound the true Rademacher complexity with the empirical Rademacher complexity, does not hold. We use the technique suggested by Shivaswamy and Jebara [22] to overcome this obstacle by introducing a set of landmark examples $U = u_1, ..., u_m$ for the regularization. Denote the function class considered by the minimal correlation optimization problem as

$$\mathbb{G}^S_{E,\lambda,\eta} := \{x \to w^T x \ : \ \frac{\lambda}{2}\|w\|^2 + \|w^T V\|^2 \leq E\} \ ,$$

where E emerges from the value of problem 1 when $w = \mathbf{0}$. Let U be the set of landmark examples, which is essentially a validation set used to calculate the correlation expressions, and does not intersect with the train set. Denote by V_U the matrix constructed similarly to the matrix V in equation 2 with the landmark examples instead of the training set examples. The function class considered when adding the landmark examples is

$$\mathbb{G}^U_{B,\lambda,\eta} := \{x \to w^T x \ : \ \frac{\lambda}{2}\|w\|^2 + \|w^T V_U\|^2 \leq B\} \ .$$

The following bound on learning the function class $\mathbb{G}^S_{E,\lambda,\eta}$ holds:

Theorem 1. *Fix $\gamma > 0$ and let the training set S be drown iid from a probability distribution D. For any $g \in \mathbb{G}^S_{E,\lambda,\eta}$, the following bound holds for $B = E + O\left(\frac{1}{\sqrt{n}}\right)$ with probability at least $1 - \delta$,*

$$Pr_D[y \neq sign(g(x))] \leq \frac{1}{m\gamma}\sum_{i=1}^{m}\xi_i + 3\sqrt{\frac{ln(8/\delta)}{2m}} + O\left(\frac{1}{\sqrt{m}\sqrt{trK}}\right)$$

$$+\frac{4\sqrt{2B}}{m\gamma}\mathbb{E}_{\mathbb{U}}\left(\sum_{i=1}^{m} x_i^T\left(\lambda I + V_U V_U^T\right)^{-1} x_i\right)^{0.5} .$$

The proof follows from Theorem 18(*iii*) in [22].

4 Experiments

We have conducted experiments on several data sets to evaluate the performance of our algorithm in comparison with other classification algorithms. Specifically, for each dataset, multiple binary problems were derived and used to compare the performance of various classifiers. To measure the preformance enhancement arising from the classifiers diversity while eliminating the effects of the ensemble, we compared our method to boosting over decision stumps and over SVM as weak classifiers, and to an ensemble of bagged SVMs [3].

The regularization parameter of SVM (λ) is determined using cross validation on the training set of each experiment. The same parameter is then used as the regularization parameter of the minimal correlation ensemble. The kernel parameters for non-linear SVM are determined similarly for the baseline classifier, and used for the entire ensemble.

The minimal correlation balancing parameters η_j are calibrated as follows: first, all η_j for $j = 1, \ldots, k-1$ are set to 1. When the optimal solution is found, the correlation between the new classifier and each of the former $k-1$ classifiers is evaluated for both classes. Let $\beta \in (0, 1)$ be a predetermined upper bound on the magnitude of each of the correlations. If the magnitude of the correlation of the currently learned classifier with classifier j over any of the classes is higher then β, then η_j is enlarged to $2\eta_j$. If any of the tradeoffs is updated, we solve the optimization problem with the new tradeoffs. This process is repeated until all correlations are bellow β. In all of our experiments we fix $\beta = 0.8$.

We train ensembles of $k = 4$ classifiers by default, and demonstrate performance as a function of this parameter in Figure 2. For learning the ensembles, the training set is split to 80% training of the base classifiers and 20% "validation" for training the ensemble classifier. For the baseline classifiers, the entire training set is used.

Letter Recognition. Taken from the UCI Machine Learning Repository [20], consists of 20,000 example images of the 26 letters of the English alphabet in the upper case. The letters are derived from 20 fonts that are randomly distorted to form black-and-white images. Each image is analyzed to produce a feature vector of 16 numerical attributes.

We derive several sets of binary problems with varying complexity. In the first set, we perform 26 one-vs-all classification experiments. The results, shown in Figure 1(a), indicate that the minimal correlation ensemble outperforms both linear and gaussian SVM. In a second set of experiments, we create composite classes by combining pairs of letters. For each of the $\binom{26}{2}$ possibilities, we create one positive class that is a union of the examples of two letters. As depicted in Figure 1(b), the gap in performance in this case is even larger in favor of the minimal correlation ensemble. The third set of experiments is similar and contains positive classes that consist of the union of three random letters, see Figure 1(c). The experiment is repeated for the Gaussian kernel, see Figure 1(j-l). The comparison to GentleBoost over SVMs shown in Figure 1(d-f). Figure 1(g-i) compares a different ensemble method based on multiple SVMs trained on

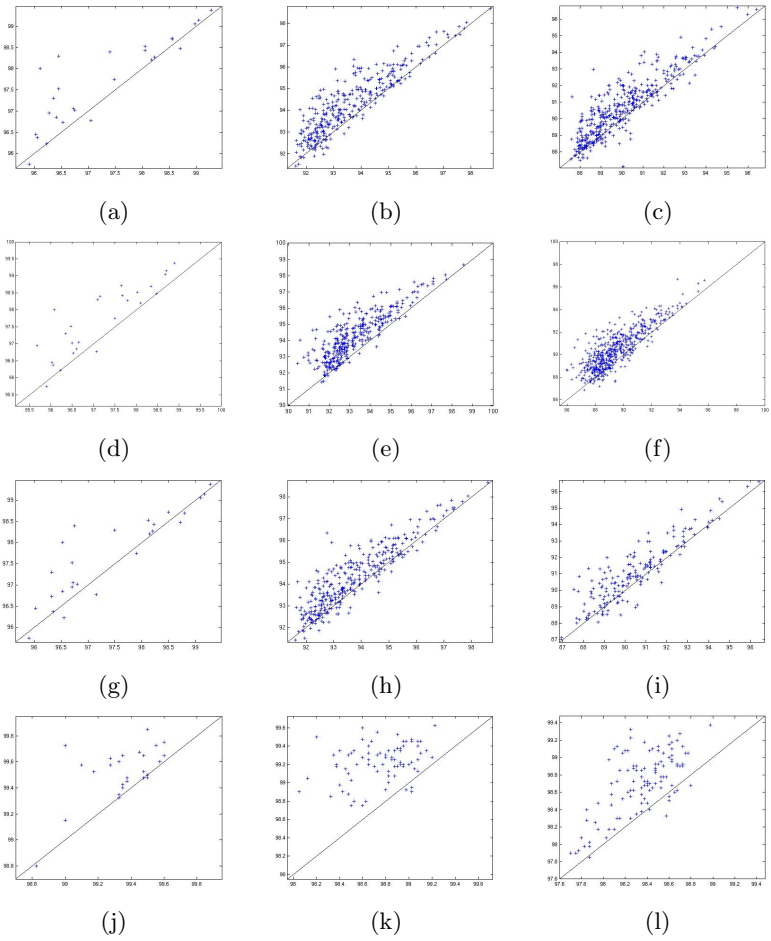


Fig. 1. Experiments on the Letter Recognition Dataset - minimal correlation ensemble vs. other classification methods. The first row (a-c) compares linear SVM to linear-based minimal correlation ensemble on one-vs-all, two-vs-all and three-vs-all binary problems from left to right respectively, the second row (d-f) compares the minimal correlation ensembles to boosting with SVM as the weak classifier on the same sets of problems. The third row (g-i) compares boosting over bagged SVMs to linear-based minimal correlation ensemble, and the forth row (j-l) compares gaussian SVM to gaussian-based minimal correlation ensemble. In all the graphs, the x axis shows the baseline classification accuracy and the y axis shows the minimal correlation ensemble classification accuracy. Every '+' represents the results of one experiment.

multiple random subsets of the training data and assembled using GentleBoost with decision stumps. Minimal correlation ensemble surpasses both methods. Finally, GentleBoost with decision stumps was applied over the originl feature space and obtained significantly lower performance levels.

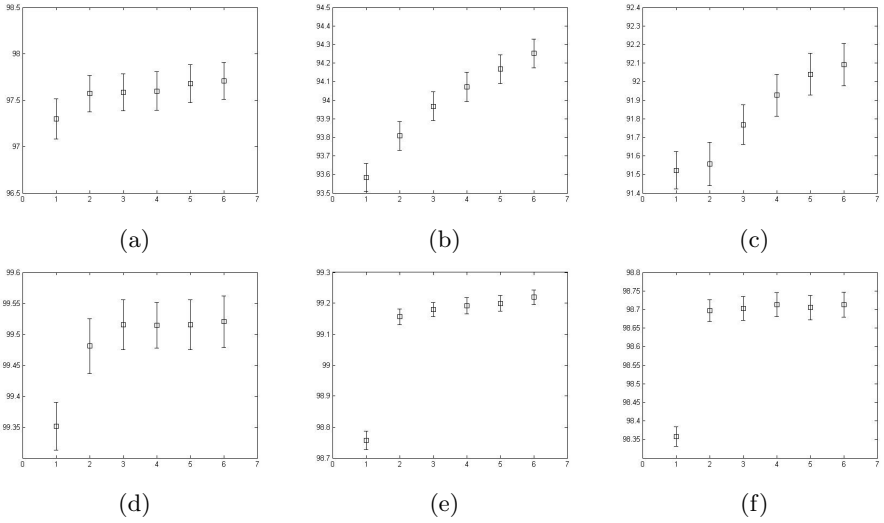


Fig. 2. Average classification accuracy as a function of the number of ensemble classifiers for one-vs-all, two-vs-all and three-vs-all classification problems on the letter data set. The x-axis indicates the number of learned classifiers; The y-axis is the obtained performance level for each sized ensemble. (a-c) Linear SVM. (d-f) Gaussian kernel SVM. Note that one classifier means the baseline SVM classifier.

To measure the impact of the number of learned classifiers, we learned for every problem an ensemble of two to six classifiers. Figure 2 demonstrates the performance improvement as a factor of the number of base classifiers for the one-vs-all, two-vs-all and three-vs-all experiments. In graphs (a)-(c) linear SVM is used as the base classifier, and it is clear that each of the sequential classifiers contributes to the performance. In graphs (d)-(f) the gaussian kernel SVM is used and it seems that in this case the third and on classifiers are not necessary.

Multi-modal Pedestrian Detection. A large dataset made available to us by a private research lab. The underlying problem consists of template-based detection in surveillance cameras, where multiple templates are used due to the large variability in the view angle. The dataset contains detections and false detections obtained with over 30 different models. The template based detection is a preliminary stage, in which for each of the pedestrian templates a separate set of detections is found. It is our task to filter out the false detections for each of these detectors. Each possible detection is characterized by different features including (a) the intensities of the detected region (900 features), (b) the coefficients of the affine transformation each model template undergoes to match the detected region (6 features), (c) the SIFT [19] descriptor of the detected

region (128 features), and (d) the HOG [5] descriptor (2048 features) of the detected region.

We have compared the performance of Linear SVM to the performance of the online version of the minimal correction ensemble (given the number of experiments needed, the batch version was not practical). For each of the four feature types, and for each of the tens of models, we record the performance of the two methods using a standard cross-validation scheme, where two thirds of the samples were used for training and one third for testing. The presented rates are averaged over 20 cross validation runs.

The results are shown in Figure 3. For the gray value based features of Figure 3(a), which are relatively weak, both methods do similarly, with the baseline methods outperforming our method on 48% of the models. However, for the more sophisticated representations, shown in Figure 3(b-d), the minimal correlation ensemble outperform the baseline method for over 70% of the experiments, showing a typical average increase in performance of about 10%.

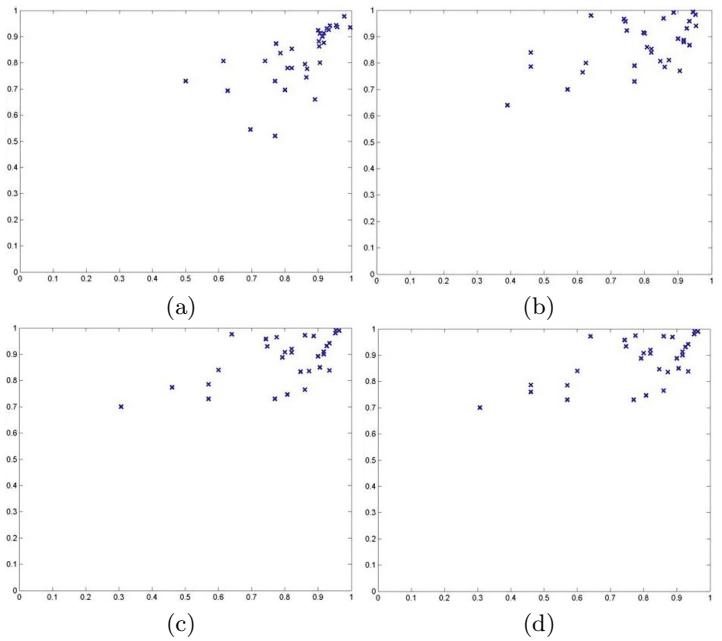
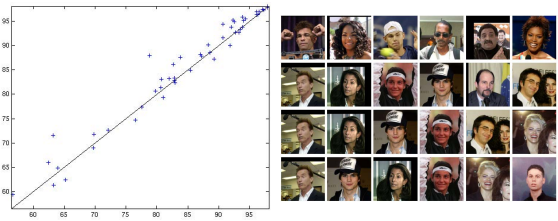


Fig. 3. The graphs present the results obtained for recognizing true detections vs. false detections on tens of template-based models. Each of the four graphs presents results for one feature type. (a) the gray values of the detected region; (b) the coefficients of the model to detected region affine transformation; (c,d) SIFT and HOG edge histograms of the found patch. Each point in each graph represents one model for which accuracy of differentiating true from false detections is measured for both the baseline SVM method (x-axis) and the minimal correlation ensemble method (y).

Facial Attributes. The first 1,000 images of the 'Labeled Faces in the Wild' dataset [13] are labeled by a variety of attributes proposed to be useful for face recognition in [16], available at [1]. Examples of such attributes include 'Blond Hair', 'Bags under eyes', 'Frowning', 'Mouth Closed' and 'Middle Aged'. For each attribute we conduct one binary experiment recognizing the presence of this feature. Each face image is automatically aligned using automatically detected feature points [8] and represented by the a histogram of LBP features [2]. Cross validation experiments (50% train; 50% test; average of 20 repeats) are performed and the results, comparing the minimum correlation ensemble with linear SVM (often used for such tasks; gaussian SVM, not shown, performs worse) are presented in Figure 4(a). See Figure 4(b) for the top detections of each of the first four minimally correlated classifiers learned for the attribute 'Eyebrow Thick'. As can be seen, these detections are not independent, demonstrating that reducing correlation differs from grouping the samples and learning multiple models or other means of strongly enforcing decorrelation.



(a) (b)

Fig. 4. Experiments on the facial attributes of the 'Labeled Faces in the Wild' dataset. (a) A comparison of the performance of linear SVM to to the linear minimal correlation ensemble for various attributes. The x axis shows the SVM performance and the y axis shows the minimal correlation ensemble performance. Every '+' represents the result on one facial attribute. (b) Highest-ranked images for the 'Eyebrow Thick' attribute. The images in the first row are the examples ranked highest by the linear SVM, the images in the rows underneath were ranked highest by the second, third and forth classifiers of the minimal correlation ensemble.

Wound Segmentation. The analysis of biological assays such as wound healing and scatter assay requires separating between multi-cellular and background regions in cellular bright field images. This is a challenging task due to the hetrogenious nature of both the foreground and the background (see Figure 6(a,b)). Automating this process would save time and effort in future studies, especially considering the high amount of data currently available. Several algorithms and tools for automatic analysis have recently been proposed to deal with this task [11,17]. To the best of our knowledge, the only freely available software for automatic analysis of wound healing that performs reasonably well on bright field images without specific parameter setting is TScratch [11].

TScratch uses fast discrete curvelet transform [4] to segment and measure the area occupied by cells in an image. The curvelet transform extracts gradient information in many scales, orientations and positions in a given image, and encodes it as curvelet coefficients. It selects two scale levels to fit the gradient details found in cells' contours, and generates a curvelet magnitude image by combining the two scale levels, which incorporates the details of the original image in the selected scales. Morphological operators are applied to refine the curvelet magnitude image. Finally, the curvelet magnitude image is partitioned into occupied and free regions using a threshold. This approach was first applied for edge detection in microscopy images [10].

We apply minimal correlation ensemble to the segmentation solution described in [24], based on learning the local appearance of cellular versus background (non-cell) small image-patches in wound healing assay images. A given image is partitioned into small patches of 20x20 pixels. From each patch, the extracted features are image and patch gray-level mean and standard deviation, patch's gray level histogram, histogram of patch's gradient intensity, histogram of patch's spatial smoothness (the difference between the intensity of a pixel and its neighborhood),and similar features from a broader area surrounding the patch. All features are concatenated to produce combined feature vectors of length 137.

The datasets, taken from [24], are: (a) 24 images available from the TScratch website (`http://chaton.inf.ethz.ch/software/`). (b) 20 images of cell populations of brain metastatic melanoma that were acquired using an inverted microscope. (c) 28 DIC images of DA3 cell lines acquired using an LSM-410 microscopemicroscope (Zeiss, Germany). (d) 54 DIC images of similar cell lines acquired using an LSM-510 microscope (Zeiss, Germany). In order to create a generic model, the training set consists of twenty arbitrary images collected at random from all data sets. The model is tested on the remaining images of each of the four data sets separately.

Figure 6(c-f) depicts the obtained performance, comparing vanilla linear SVM to the proposed Minimal Correlation and to TScratch's results (we used the continuous output of TScratch and not a single decision in order to plot ROC curves). As can be seen, the patch classification method implemented here outperforms TScratch in all datasets, and the proposed method significantly outperforms SVM in three out of four data sets. Table 5 compares the Area under curve of TScratch, linear SVM and minimal correlation ensemble for all datasets.

	Tscratch data	Melanoma	DA3 (LSM-410)	DA3 (LSM-510)
Minimal Correlation	96.6	94.1	95.6	95.5
Tscratch algo.	93.4	84	94.4	94.9
Boosted SVM	95.9	94.3	95.5	96.6
Bagging	95.7	92.6	95.4	95.4

Fig. 5. A comparison of the Area under curve (AUC) of linear SVM vs. linear minimal correlation ensemble with two to six classifiers over the four wound healing datasets

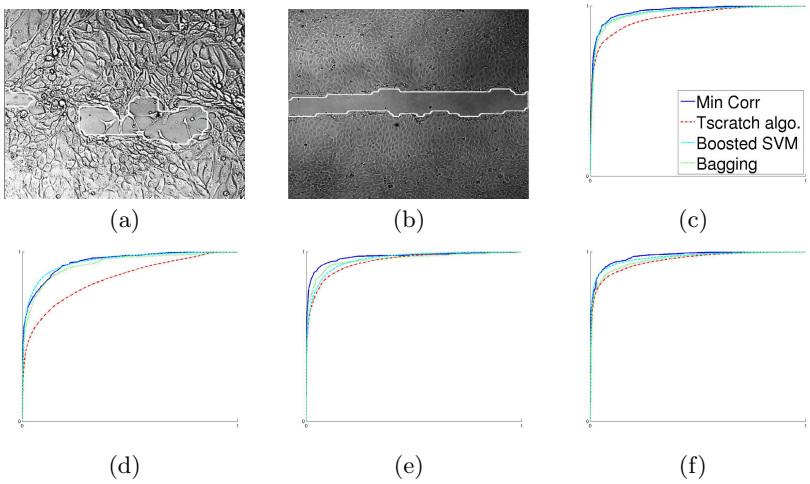


Fig. 6. (a,b) Cell foreground segmentation examples. (c-f) Comparison of ROC curves on four datasets of wound healing images. The minimal correlation ensemble (solid blue line) is compared to the Tscratch (red dash-dot) algorithm, the Boosted SVM (dashed cyan) and to Bagging (dotted green). The X-axis is the false positive rate abd the Y-axis is the true positive rate. (c) the Tscratch dataset, (d) Melanoma dataset, (e) and (f) DA3 cell lines acquired with two different microscopes.

5 Summary

We employ a variant of SVM that learns multiple classifiers that are especially suited for combination in an ensemble. The optimization is done efficiently using QP, and we also present a kernelized version as well as an efficient online version. Experiments on a variety of visual domains demonstrate the effectiveness of the proposed method. It is shown that the method is especially effective on compound classes containing several sub-classes and that the results are stable with respect to the number of employed base classifiers. It is also demonstrated that using the minimal correlation principle is not the same as learning several classifiers on different subsets of each class.

Acknowledgments. This research was supported by the I-CORE Program of the Planning and Budgeting Committee and The Israel Science Foundation (grant No. 4/11). The research was carried out in partial fulfillment of the requirements for the Ph.D. degree of Noga Levy.

References

1. `http://www.cs.columbia.edu/CAVE/databases/pubfig/download/lfw_attributes.txt`
2. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. PAMI 28(12), 2037–2041 (2006)

3. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
4. Candes, E., Demanet, L., Donoho, D., Ying, L.: Fast discrete curvelet transforms. In: Multiscale Modeling and Simulation (2006)
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
6. Dietterich, T.G.: Ensemble Methods in Machine Learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
7. Dzeroski, S., Zenko, B.: Is combining classifiers with stacking better than selecting the best one? Machine Learning 54(3), 255–273 (2004)
8. Everingham, M., Sivic, J., Zisserman, A.: "Hello! My name is... Buffy" – automatic naming of characters in TV video. In: BMVC (2006)
9. Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic Regression: a Statistical View of Boosting. The Annals of Statistics 38(2) (2000)
10. Geback, T., Koumoutsakos, P.: Edge detection in microscopy images using curvelets. BMC Bioinformatics 10(75) (2009)
11. Geback, T., Schulz, M., Koumoutsakos, P., Detmar, M.: Tscratch: a novel and simple software tool for automated analysis of monolayer wound healing assays. Biotechniques 46, 265–274 (2009)
12. Hu, G., Mao, Z.: Bagging ensemble of svm based on negative correlation learning. In: IEEE International Conference on ICIS 2009, vol. 1, pp. 279–283 (2009)
13. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. University of Massachusetts, Amherst, TR 07-49 (October 2007)
14. Kim, H.C., Pang, S., Je, H.M., Kim, D., Bang, S.Y.: Constructing support vector machine ensemble. Pattern Recognition 36(12), 2757–2767 (2003)
15. Kocsor, A., Kovács, K., Szepesvári, C.: Margin Maximizing Discriminant Analysis. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 227–238. Springer, Heidelberg (2004)
16. Kumar, N., Berg, A.C., Belhumeur, P.N., Nayar, S.K.: Attribute and simile classifiers for face verification. In: ICCV, pp. 365–372 (2009)
17. Lamprecht, M., Sabatini, D., Carpenter, A.: Cellprofiler: free, versatile software for automated biological image analysis. Biotechniques 42, 71–75 (2007)
18. Liu, Y., Yao, X.: Simultaneous training of negatively correlated neural networks in an ensemble. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 29, 716–725 (1999)
19. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
20. Murphy, P., Aha, D.: UCI Repository of machine learning databases. Tech. rep., U. California, Dept. of Information and Computer Science, CA, US (1994)
21. Shalev-Shwartz, S., Singer, Y., Srebro, N.: Pegasos: Primal estimated sub-gradient solver for svm. In: ICML (2007)
22. Shivaswamy, P.K., Jebara, T.: Maximum relative margin and data-dependent regularization. Journal of Machine Learning Research (2010)
23. Wolpert, D.H.: Stacked generalization. Neural Networks 5(2), 241–259 (1992)
24. Zaritsky, A., Natan, S., Horev, J., Hecht, I., Wolf, L., Ben-Jacob, E., Tsarfaty, I.: Cell motility dynamics: A novel segmentation algorithm to quantify multi-cellular bright field microscopy images. PLoS ONE 6 (2011)