AUC Maximizing Support Vector Learning

Brefeld, Ulf; Scheffer, Tobias

Published in:
ROC Analysis in Machine Learning

Publication date:
2005

Document Version
Peer reviewed version

AUC Maximizing Support Vector Learning

Ulf Brefeld BREFELD@INFORMATIK.HU-BERLIN.DE
Tobias Scheffer SCHEFFER@INFORMATIK.HU-BERLIN.DE
Humboldt-Universität zu Berlin, Department of Computer Science, Unter den Linden 6, 10099 Berlin,Germany

Abstract

The area under the ROC curve (AUC) is a natural performance measure when the goal is to find a discriminative decision function. We present a rigorous derivation of an AUC maximizing Support Vector Machine; its optimization criterion is composed of a convex bound on the AUC and a margin term. The number of constraints in the optimization problem grows quadratically in the number of examples. We discuss an approximation for large data sets that clusters the constraints. Our experiments show that the AUC maximizing Support Vector Machine does in fact lead to higher AUC values.

1. Introduction

Receiver Operating Characteristics (ROC) analysis is now being recognized as a useful tool for machine learning because it allows to assess uncalibrated decision functions, even when the prior distribution of the classes is not known (Provost et al., 1998; Bradley, 1997). A decision function can be compared against a threshold to yield a classifier. The ROC curve details the rate of true positives against false positives over the range of possible threshold values. The area of that curve is the probability that a randomly drawn positive example has a higher decision function value than a random negative example; it is called the *AUC (area under ROC curve)*.

When the goal of a learning problem is to find a decision function with high AUC value, then it is natural to use a learning algorithm that directly maximizes this criterion. Over the last years, AUC maximizing versions of several learning algorithms have been developed. We contribute to the field by characterizing an AUC maximizing Support Vector Machine (SVM).

We construct an appropriate convex optimization criterion and derive corresponding dual quadratic programs that can be solved by standard QP solvers. We study empirically whether the AUC maximizing SVM optimizes the AUC more effectively than the regular SVM. Our experiments cover linear and polynomial kernels. The number of constraints and parameters in this problem grows quadratically in the sample size. This increases the relative contribution of the slack terms to the optimization criterion; we study how this influences the optimal value of the regularization parameter. For large samples, the optimization problem becomes hard. We approximate the optimization problem by clustering the resulting constraints.

The rest of our paper is structured as follows. We introduce the problem setting in Section 2 and derive the AUC maximizing SVM in Section 3. In Section 4, we apply clustering methods to reduce the number of constraints in the optimization problem. We present our experiments in Section 5 and discuss related work in Section 6. Section 7 concludes.

2. Preliminaries

The ROC curve of a decision function f characterizes every possible trade-off between false positives and true positives that can be achieved by comparing $f(x)$ to a threshold. The ROC curve plots the number of true positives on the y-axis against the number of false positives on the x-axis. The area under the ROC curve – the AUC – is equal to the probability that the decision function f assigns a higher value to a randomly drawn positive example x^+ than to a randomly drawn negative example x^-,

$$AUC(f) = Pr\left(f(x^+) > f(x^-)\right). \quad (1)$$

The AUC refers to the true distribution of positive and negative instances, but it can be estimated using a sample. The normalized Wilcoxon-Mann-Whitney statistic (Yan et al., 2003) gives the maximum likeli-

Appearing in *Proceedings of the ICML 2005 workshop on ROC Analysis in Machine Learning*, Bonn, Germany, 2005.

hood estimate of the true AUC (Equation 1) given n^+ positive and n^- negative examples:

$$\widehat{AUC}(f) = \frac{\sum_{i=1}^{n^+}\sum_{j=1}^{n^-} 1_{f(x_i^+)>f(x_j^-)}}{n^+n^-}. \quad (2)$$

The two sums in Equation (2) iterate over all pairs of positive and negative examples. Each pair that satisfies $f(x^+) > f(x^-)$ contributes with $1/(n^+n^-)$ to the overall AUC performance. Maximizing the AUC is therefore equivalent to maximizing the number of pairs satisfying $f(x^+) > f(x^-)$.

The AUC is invariant with respect to the *a priori* class distribution; it is also independent of decision thresholds because if $f(x) > f(x')$, then $f(x)+c > f(x')+c$ for any c. In the remainder of this paper we analyze AUC maximization with the unthresholded model

$$f(x) = \langle w, \phi(x)\rangle. \quad (3)$$

The transformation $\phi : \mathcal{X} \mapsto \mathcal{F}$ denotes an implicit preprocessing of the input data which is equivalent to mapping the input space $\mathcal{X}$ into some feature space $\mathcal{F}$. However, no explicit representation of ϕ or $\mathcal{F}$ is required. Like for all other kernel-based learning algorithms, direct computation of the inner product in $\mathcal{F}$ by a kernel function $k(x, x') = \langle\phi(x), \phi(x')\rangle$ suffices to learn in feature space $\mathcal{F}$. We will refer to k as matrix with elements $k_{ij} = k(x_i, x_j)$.

3. AUC Support Vector Learning

In this section we present the AUC maximizing SVM (AUC SVM). We first derive the optimization problem whose objective function consists of an upper bound of the number of examples that violate the constraint $f(x^+) > f(x^-)$ and a margin based complexity term. In a second step we present the complete derivation of the AUC maximizing SVM with its primal and dual Lagrangians. We will characterize the relationship between linear AUC SVM and "vanilla" SVM.

The key concept of AUC support vector learning is based on the observation that AUC performance depends directly on the number of pairs (x_i^+, x_j^-), $i = 1, \ldots, n^+$ and $j = 1, \ldots, n^-$, that satisfy $f(x_i^+) > f(x_j^-)$ or, equivalently,

$$\langle w, \phi(x_i^+)\rangle - \langle w, \phi(x_j^-)\rangle > 0. \quad (4)$$

We introduce a margin $\gamma = \bar{\gamma}/\|w\|$ as confidence measure of an actual solution into Equation (4) and obtain the hard margin AUC Optimization Problem 1.

Optimization Problem 1 *Given n^+ positive and n^- negative examples; maximize γ subject to the constraints $\forall_{i=1}^{n^+}\forall_{j=1}^{n^-} \langle w, \phi(x_i^+)\rangle - \langle w, \phi(x_j^-)\rangle \geq \bar{\gamma}$.*

The geometrical margin γ is maximized by either fixing $\|w\| = const$ and maximizing the functional margin $\bar{\gamma}$ or fixing $\bar{\gamma} = const$ and minimizing the norm of w. In analogy to most kernel based methods we apply the latter and set $\bar{\gamma} = 1$.

A solution to the hard margin Optimization Problem 1 but also any solution to the regular hard margin SVM optimization problem always satisfies AUC=1 on the training data because Equation (4) is always true when $\min_i f(x_i^+) - \max_j f(x_j^-) \geq \bar{\gamma}$. In general we have to allow pairwise relaxations of the margin by nonnegative slack variables ξ_{ij}, leading to the soft margin Optimization Problem 2.

Optimization Problem 2 *Given n^+ positive and n^- negative examples, let $C > 0$ and $r = 1, 2$; over all w and ξ, minimize $\frac{1}{2}\|w\|^2 + \frac{C}{r}\sum \xi_{ij}^r$ subject to the constraints $\forall_{i=1}^{n^+}\forall_{j=1}^{n^-} \langle w, \phi(x_i^+)\rangle - \langle w, \phi(x_j^-)\rangle \geq 1 - \xi_{ij}$ and $\forall_{i=1}^{n^+}\forall_{j=1}^{n^-} \xi_{ij} \geq 0$.*

Optimization Problem 2 is convex and for $r = 1, 2$ also quadratic because of its quadratic objective and linear constraints. The regularization constant C determines the trade-off between the complexity term and the sum of the slacks. According to Theorem 1 the latter may be interpreted as an upper bound of the number of pairs that do not satisfy Equation (4).

Theorem 1 *Given Optimization Problem 2, the sum $\sum \xi_{ij}^r$ with $r = 1, 2$ upper-bounds the number of pairs (x_i^+, x_j^-) in the sample that violate $f(x_i^+) > f(x_j^-)$.*

Proof. The first set of constraints of Optimization Problem 2 guarantees Equation (5). For any pair that violates $f(x_i^+) > f(x_j^-)$, Equation (7) follows and the pair contributes at least 1 to the sum $\sum_{ij} \xi_{ij}^r$.

$$\langle w, \phi(x_i^+)\rangle - \langle w, \phi(x_j^-)\rangle \geq 1 - \xi_{ij} \quad (5)$$

$$\Leftrightarrow \quad \xi_{ij} \geq 1 - (\langle w, \phi(x_i^+)\rangle - \langle w, \phi(x_j^-)\rangle) \quad (6)$$

$$\Leftrightarrow \quad \xi_{ij} \geq 1 \text{ if } f(x_i^+) - f(x_j^-) \leq 0 \quad (7)$$

All ξ_{ij} are nonnegative (guaranteed by the nonnegativity constraints), and therefore $\sum_{ij} \xi_{ij}^r$ with $r = 1, 2$ is greater than or equal to the number of pairs in violation of $f(x_i^+) > f(x_j^-)$. ∎

The constraints of Optimization Problem 2 can be integrated into the objective function by introducing a Lagrange multiplier for each constraint. In the following we describe the derivation of the 1-norm and 2-norm AUC maximizing SVM, respectively.

1-Norm AUC SVM

For the 1-norm AUC maximizing SVM we transform

Optimization Problem 2 for $r = 1$ into the dual representation and resolve the primal Lagrangian

$$L_p^1(w, \xi, \alpha, \beta) = \frac{1}{2}\|w\|^2 + C\sum_{ij}\xi_{ij} - \sum_{ij}\beta_{ij}\xi_{ij} - \sum_{ij}\alpha_{ij}\left(\langle w, \phi(x_i)\rangle - \langle w, \phi(x_j)\rangle - 1 + \xi_{ij}\right).$$

Since Optimization Problem 2 is convex the Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient for w and ξ to be a solution. The KKT conditions are given by

$$\begin{aligned}
\frac{\partial L_p^1}{\partial w} = w - \sum_{ij}\alpha_{ij}\left(\phi(x_i) - \phi(x_j)\right) &= \mathbf{0} && (8)\\
\frac{\partial L_p^1}{\partial \xi_{ij}} = C - \alpha_{ij} - \beta_{ij} &= 0 && (9)\\
\langle w, \phi(x_i)\rangle - \langle w, \phi(x_j)\rangle - 1 + \xi_{ij} &\geq 0\\
\xi_{ij} &\geq 0\\
\alpha_{ij} &\geq 0\\
\beta_{ij} &\geq 0 && (10)\\
\alpha_{ij}\left[\langle w, \phi(x_i)\rangle - \langle w, \phi(x_j)\rangle - 1 + \xi_{ij}\right] &= 0 && (11)\\
\beta_{ij}\,\xi_{ij} &= 0.
\end{aligned}$$

The substitution of Equation (8) and (9) into the primal Lagrangian removes its dependence on the primal variables and we resolve the corresponding dual

$$\begin{aligned}
L_d^1(\alpha) &= -\tfrac{1}{2}\sum_{ij,uv}\alpha_{ij}\alpha_{uv}\left(\phi(x_i) - \phi(x_j)\right)\left(\phi(x_u) - \phi(x_v)\right) + \sum_{ij}\alpha_{ij}\\
&= \sum_{ij}\alpha_{ij} - \tfrac{1}{2}\sum_{ij,uv}\alpha_{ij}\alpha_{uv}\left(k_{iu} - k_{iv} - k_{ju} + k_{jv}\right).
\end{aligned}$$

Equations (9) and (10) enforce $\alpha_{ij} < C$. Although the derivative of ξ_{ij} does not occur in the dual, it bounds α_{ij} implicitly within the box $0 \leq \alpha_{ij} \leq C$. We define the kernel $K' = \{k'_{ij,uv}\}$ with $k'_{ij,uv} = k_{iu} - k_{iv} - k_{ju} + k_{jv}$ and derive the 1-norm AUC maximizing support vector optimization problem.

Optimization Problem 3 *Given n^+ positive and n^- negative examples; over all α_{ij}, maximize $\sum_{ij}\alpha_{ij} - \frac{1}{2}\sum_{ij,uv}\alpha_{ij}\alpha_{uv}\,k'_{ij,uv}$ subject to the constraints $\forall_{i=1}^{n^+}\forall_{j=1}^{n^-}\, 0 \leq \alpha_{ij} < C$.*

Optimization problem 3 paves the way to an implementation of the AUC maximizing 1-norm SVM. It can be solved by a standard QP solver.

The KKT complementarity conditions in Equation (11) guarantee sparsity in the optimal α_{ij}^* because $\langle w, \phi(x_i)\rangle - \langle w, \phi(x_j)\rangle - 1 + \xi_{ij} \neq 0$ enforces $\alpha_{ij}^* = 0$. Thus, all example pairs (x_i^+, x_j^-) for which the corresponding $\alpha_{ij}^* \neq 0$ holds are support vectors and the primal solution w^* can be written as their linear combination

$$w^* = \sum_{ij:\alpha_{ij}^*\neq 0}\alpha_{ij}^*(\phi(x_i) - \phi(x_j)).$$

The solution has a geometric margin of $\gamma = \|w^*\|^{-1}$.

2-Norm AUC SVM

For $r = 2$ we may drop the nonnegativity constraints $\xi_{ij} \geq 0$ in Optimization Problem 2 since $\xi_{ij} < 0$ satisfies $\langle w, \phi(x_i)\rangle - \langle w, \phi(x_j)\rangle \geq 1 - \xi_{ij}$ and $\sum\xi_{ij}^2$ guarantees the objective to be positive. The primal Lagrangian is then given by

$$L_p^2(w, \xi, \alpha) = \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_{ij}\xi_{ij}^2 - \sum_{ij}\alpha_{ij}\left(\langle w, \phi(x_i)\rangle - \langle w, \phi(x_j)\rangle - 1 + \xi_{ij}\right).$$

We omit the complete presentation of the KKT conditions which are similar to the 1-norm case and proceed directly to the optimal primal variables,

$$\begin{aligned}
w^* &= \sum_{ij}\alpha_{ij}\left(\phi(x_i) - \phi(x_j)\right)\\
\xi_{ij}^* &= \frac{1}{C}\,\alpha_{ij}.
\end{aligned}$$

Again, we derive the corresponding dual by a substitution of the primal variables. The dual is given by

$$\begin{aligned}
L_d^2(\alpha) &= -\frac{1}{2}\sum_{ij,uv}\alpha_{ij}\alpha_{uv}\left(k_{iu} - k_{iv} - k_{ju} + k_{jv}\right)\\
&\quad + \sum_{ij}\alpha_{ij} - \frac{1}{2C}\sum_{ij}\alpha_{ij}^2. \qquad (12)
\end{aligned}$$

The last summand of Equation (12) may be integrated into the kernel by $K'' = K' + \frac{1}{C}\mathbf{1}$, where $\mathbf{1}$ denotes the identity. The 2-norm AUC maximizing support vector optimization problem is stated in Optimization Problem 4.

Optimization Problem 4 *Given n^+ postive and n^- negative examples; over all α_{ij} maximize $\sum_{ij}\alpha_{ij} - \frac{1}{2}\sum_{ij,uv}\alpha_{ij}\alpha_{uv}\,k''_{ij,uv}$ subject to the constraints $\forall_{i=1}^{n^+}\forall_{j=1}^{n^-}\,\alpha_{ij} \geq 0$.*

Optimization Problem 4 shows how the AUC maximizing 2-norm SVM can be implemented using a standard QP solver. The resulting decision function can be written in dual coordinates as

$$\begin{aligned}
f(x) &= \langle w^*, \phi(x)\rangle\\
&= \sum_{ij:\alpha_{ij}^*\neq 0}\alpha_{ij}^*\left(k(x_i, x) - k(x_j, x)\right).
\end{aligned}$$

Analogy to the regular SVM
Theorem 2 characterizes the relation between "vanilla" SVM and AUC maximizing support vector learning for the case of a linear kernel. Intuitively, the linear AUC maximizing SVM can be "emulated" by a regular SVM without threshold when a new training set is constructed that consists of n^+n^- positive examples $z_{ij} = x_i^+ - x_j^-$. This analogy does not hold for nonlinear kernels.

Theorem 2 *In case of a linear kernel $k(x, x') = \langle x, x' \rangle$ the optimization problem of an unthresholded one class SVM with training examples $(z_{11}, +1), \ldots, (z_{n^+n^-}, +1)$ where $\forall_{i=1}^{n^+} \forall_{j=1}^{n^-} z_{ij} = x_i^+ - x_j^-$ is equivalent to Optimization Problem 2.*

Proof. The soft margin optimization problem of the SVM is given by

$$\begin{array}{lrcl} \min\limits_{w,\xi} & \frac{1}{2}\|w\|^2 + \frac{C}{r}\sum \xi_{ij}^r & & \\ \text{s.t.} & y_{ij}(\langle w, z_{ij}\rangle + b) & \geq & 1 - \xi_{ij} \\ & \xi_{ij} & \geq & 0, \end{array}$$

for $1 \leq i \leq n^+$ and $1 \leq j \leq n^-$. The substitution of $z_{ij} = x_i^+ - x_j^-$, $b = 0$ and $y_{ij} = 1$ leads to the optimization problem

$$\begin{array}{lrcl} \min\limits_{w,\xi} & \frac{1}{2}\|w\|^2 + \frac{C}{r}\sum \xi_{ij}^r & & \\ \text{s.t.} & \langle w, x_i^+ - x_j^-\rangle & \geq & 1 - \xi_{ij} \\ & \xi_{ij} & \geq & 0, \end{array}$$

for $1 \leq i \leq n^+$ and $1 \leq j \leq n^-$. The choice of $k = \langle \cdot, \cdot \rangle$ implies $\phi(x) = x$. Therefore,

$$\begin{aligned} \langle w, x_i^+ - x_j^-\rangle &= \langle w, x_i^+\rangle - \langle w, x_j^-\rangle \\ &= \langle w, \phi(x_i^+)\rangle - \langle w, \phi(x_j^-)\rangle \end{aligned}$$

which is exactly Optimization Problem 2. ∎

The linear AUC SVM has a simple geometric interpretation. Its primal weight vector w is the normal of a hyperplane that passes through the origin. The training procedure adjusts the hyperplane such that the margin to the vectors $z_{ij} = x_i^+ - x_j^-$ is maximized.

Implementation and Execution Time
Optimization Problems 3 and 4 can be solved using a standard QP solver. Theorem 2 reveals the main disadvantage not only of the AUC maximizing SVM but of all methods whose criterion is based on the Wilcoxon-Mann-Whitney statistic. The number of constraints and parameters in the optimization problem grows quadratically in the number of examples.

The execution time of SVM training is estimated to be in $\mathcal{O}(n^2)$. Since the number of constraints and parameters grows quadratically in the number of examples n, training the AUC SVM incurs an execution time of roughly $\mathcal{O}(n^4)$. This execution time is feasible for small samples but unsatisfactory for large databases. Therefore, we discuss how the number of constraints can be reduced by clustering in Section 4.

Table 1. Approximate linear AUC maximizing SVM.

Input: n^+ positive and n^- negative examples, trade-off parameter C.

1. Generate all $z_{ij} = x_i^+ - x_j^-$. This results in (n^+n^-) vectors.
2. Clustering: Employ a linear clustering procedure to find $n^+ + n^-$ clusters. Let c_u be the cluster centers.
3. Define kernel matrix as $k_{u,v} = \langle c_u, c_v \rangle$.
4. Solve Optimization Problem 3 for 1-norm SVM or 4 for 2-norm SVM; *i.e.,* learn unthresholded one-class SVM using the c_u as positive examples.

Return decision function.

4. Approximate AUC Maximization

An approximate execution time of $\mathcal{O}(n^4)$ is acceptable for small samples but unsatisfactory for large databases. We will now study a method that reduces the number of constraints by clustering. The method requires a linear kernel. In order to achieve an overall complexity of $\mathcal{O}(n^2)$ we have to reduce the number of constraints and parameters from n^+n^- to $n^+ + n^-$.

In primal coordinates, we have one constraint for each pair of instances (Equation 13). In the linear case, this simplifies to Equation 14.

$$\langle w, \phi(x_i^+)\rangle - \langle w, \phi(x_j^-)\rangle \geq 1 - \xi_{ij} \qquad (13)$$

$$= \langle w, x_i^+ - x_j^-\rangle \geq 1 - \xi_{ij} \qquad (14)$$

Our strategy for approximating this problem is to represent the n^+n^- many pairs $x_i^+ - x_j^-$ by only $n^+ + n^-$ cluster centers and thereby reduce the number of constraints and parameters. Table 1 details this method. The execution time is $\mathcal{O}(n^2)$ for the generation of all pairs, at most $\mathcal{O}(n^2)$ for a clustering method and $\mathcal{O}(n^2)$ for the consecutive optimization procedure of the AUC SVM. Thus, the overall execution time is back in $\mathcal{O}(n^2)$. The method is only feasible for linear kernels because we calculate differences of instances; this requires explicit representation of $\phi(x)$.

The k-means algorithm meets these requirements. It

can be implemented such that it converges after usually one or very few scans of the data set (Goswami et al., 2004). Their Fast and Exact k-Means (FEKM) algorithm estimates initial cluster centers on a small sample. Starting from these approximate clusters, the FEKM algorithm then computes exact cluster centers using one or very few passes over the entire data. "Exact" here refers to those cluster centers that would be the result of the regular k-means algorithm.

5. Empirical Studies

We study which of the regular and AUC maximizing Support Vector Machine maximizes the AUC more effectively. We investigate the execution time and explore the benefit of the k-means AUC SVM.

We focus on problem domains for which the SVM is known to be a good solution and the AUC is a frequently used evaluation metric. We select text classification (the Reuters-21578 corpus) for the linear SVM and hand-written character recognition (the MNIST benchmark data set) for the polynomial kernel.

Our experimental setting is as follows. We use a variation of bootstrapping that allows us to vary the training sample size. We separate each of the six most frequent classes of the Reuters corpus from their complementary class. We draw a specified number of training examples and 1000 distinct holdout examples without replacement at random in each iteration. We average the performance on the holdout set over 100 iterations.

We discriminate hand-written digits 4 and 9 of the MNIST data set because they are the most similar pair of characters. We draw 50 examples from the training part and 500 holdout examples from the testing part that was written by distinct authors. We average 100 iterations. Error bars indicate the standard error. Our QP solver is the Loqo implementation by Alex Smola for the nonlinear and SVM^{light} for the linear case.

Does the AUC SVM maximize the AUC better than the regular SVM, and how should C be adjusted? We explore the space of values for parameter C using 25, 50, 100, 250, and 500 training examples. Figure 1 shows the results for Reuters using 100 examples, Figure 2 for MNIST using 50 training instances. Figure 4 summarizes the parameter value that maximizes the average AUC over all studied Reuters problems, for all studied sample sizes. The AUC SVM requires C between 0.0001 and 0.001, the regular SVM roughly between 0.01 and 0.1.

For each problem and each studied sample size we fix the apparently optimal C for both methods and re-

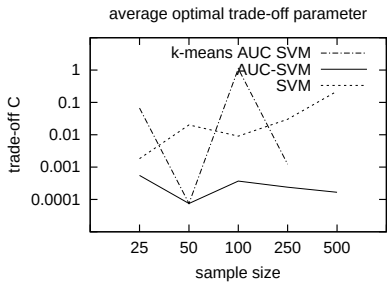

Figure 4. Optimal trade-off parameter averaged over all Reuters problems.

evaluate the AUC for these parameter settings by averaging 100 iterations with distinct resampled training and holdout samples. We compare equally many parameter values for regular and AUC maximizing SVM. Table 2 compares the resulting AUC values for SVM and regular SVM for 100 examples. In two out of six cases, we can reject the null hypothesis that both methods perform equally well at a confidence level of $\alpha = 0.05$ in favor of the AUC SVM.

Figure 3 details the AUC of regular and AUC maximizing SVM with optimized trade-off parameter for various sample sizes. For all problems and sample sizes, we conduct a two-sided test at a 5% confidence level. There is one single case in which the regular SVM is significantly better; in 9 out of 30 comparisons the AUC SVM performs significantly better than the SVM.

Table 2. AUC for Reuters; 100 examples, optimal C.

	AUC SVM	SVM
aquisition	$88.51\% \pm 0.27\%$	$88.17\% \pm 0.24\%$
crude	$87.71\% \pm 0.63\%$	$89.18\% \pm 0.48\%$
earn	$94.40\% \pm 0.13\%$	$94.15\% \pm 0.14\%$
grain	$89.71\% \pm 0.69\%$	$88.60\% \pm 0.61\%$
interest	$\mathbf{86.20\% \pm 0.68\%}$	$84.68\% \pm 0.72\%$
money-fx	$\mathbf{89.89\% \pm 0.46\%}$	$87.77\% \pm 0.53\%$

Figure 2 compares the AUC for MNIST using a polynomial kernel of degree $d = 2, 3, 4$ and 50 training examples. Table 3 details the observed AUC values. The AUC SVM appears to perform better, but the results are not significant.

From these experiments with the Reuters and MNIST problems, we conclude that, on average, the AUC SVM achieves a higher AUC than the regular SVM. In addition, the AUC SVM requires the trade-off parameter to be adjusted to roughly 1% of the optimal value for the SVM.

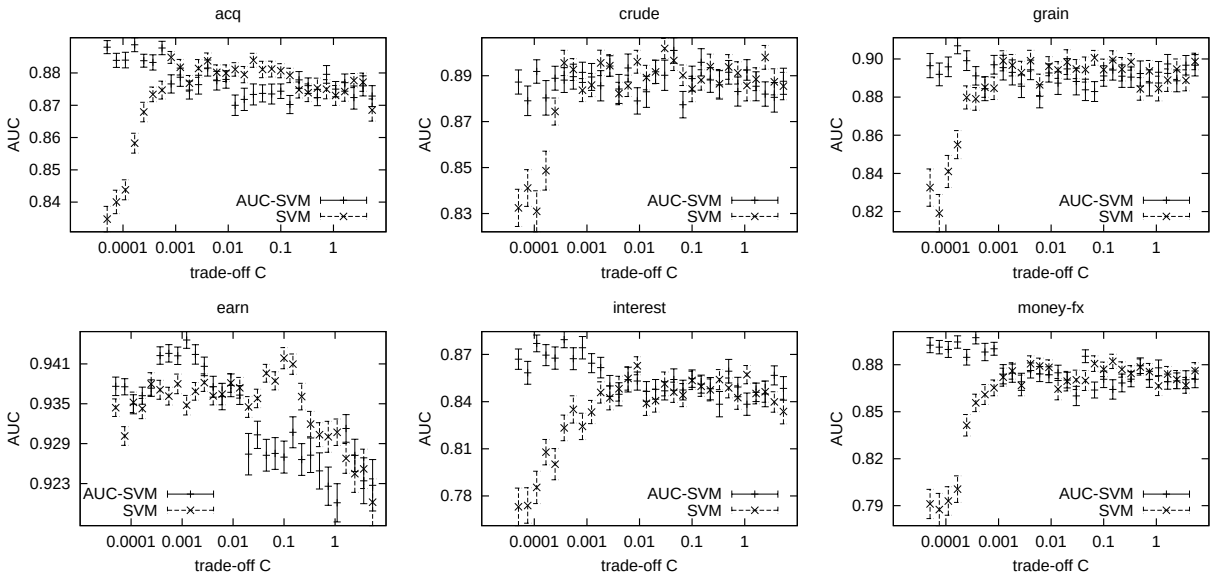

Figure 1. Exploration of trade-off values C for Reuters, 100 examples.

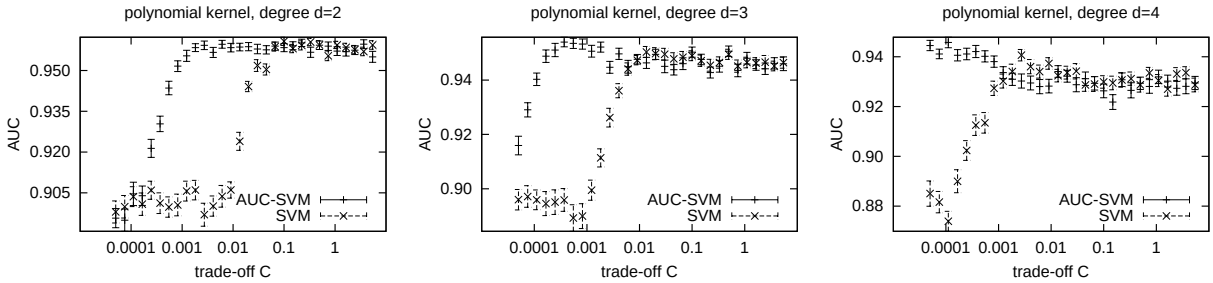

Figure 2. Exploration of trade-off values C for MNIST "4 vs. 9", 50 examples.

Table 3. AUC for MNIST; 50 examples, optimal C.

	AUC SVM	SVM
$d = 2$	$95.80\% \pm 0.18\%$	$95.52\% \pm 0.18\%$
$d = 3$	$95.42\% \pm 0.18\%$	$95.06\% \pm 0.21\%$
$d = 4$	$93.90\% \pm 0.24\%$	$93.47\% \pm 0.26\%$

How effective is the approximate optimization based on clustering? Figure 3 compares regular SVM, AUC SVM, and approximate AUC SVM using k-means clustering. We conduct two-sided tests at a 5% level and compare the k-means AUC SVM to the regular SVM. The k-means SVM beats the regular SVM significantly in 4 out of 24 cases, the regular SVM is not significantly better in any case. We conclude that, on average, the k-means AUC SVM achieves a higher AUC than the regular SVM. In addition, Figure 4 suggests that the optimal regularization parameter lies between that of regular and AUC SVM.

How do the execution time of AUC SVM and k-means AUC SVM compare to regular SVM? Figure 5 compares the execution time of regular SVM, AUC SVM, and approximate AUC SVM using fast k-means with one pass over the data. The figure shows the averaged execution time over four iterations with up to 1000 examples and fitted polynomials. The AUC SVM is slow for large samples, the fitted curve has an n^4 term. The k-means approximation is substantially faster, the fitted curve is quadratic in the sample size n. Although the regular SVM is quadratic in the sample size (computing the kernel matrix is already quadratic) the linear term dominates the observed curve. The regular SVM is faster than even the k-means AUC SVM with only one single pass over the data.

6. Related Work

ROC analysis is widely used in radar technology (Egan, 1975), psychology (Swets, 1996), medicine

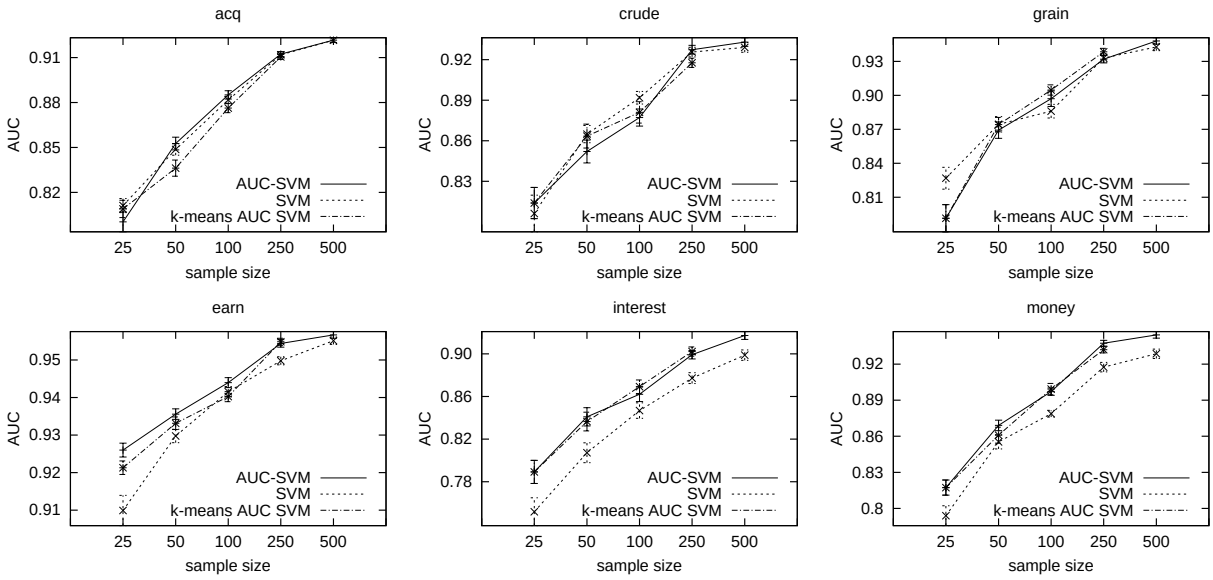

Figure 3. AUC SVM vs. regular SVM for Reuters; optimal trade-off value C.

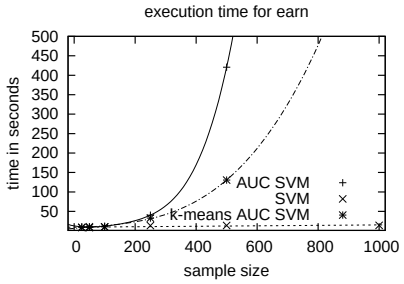

Figure 5. Execution time.

(Lusted, 1971), pattern recognition (Bradley, 1997), and, more recently, in machine learning (Provost et al., 1998). Provost and Fawcett (2001) argue that ROC analysis is useful, for instance, when class-specific costs of misclassification are not known, or the class distribution is unknown or skewed. ROC analysis is helpful to visualize, and understand the relationship between, various possible evaluation metrics (Flach, 2003, Fürnkranz & Flach, 2005).

When the learning goal is to maximize the AUC performance, it appears more appropriate to employ algorithms that directly maximize this criterion, rather than, for instance, the error rate. Therefore, AUC maximizing variants of almost all learning methods have been developed, including decision trees (Ferri et al., 2002), rules (Fawcett, 2001), boosting (Freund et al., 1998, Cortes & Mohri, 2003), logistic regression (Herschtal & Raskutti, 2004) and subgroup discovery (Kavšek et al., 2004).

In the light of previous work, an AUC maximizing kernel classifier is overdue. In this paper, we present the first rigorous derivation of an AUC SVM that in fact maximizes the AUC more effectively than the SVM. Previous efforts to implement an AUC maximizing kernel classifier (Rakotomamonjy, 2004) have so far only lead to methods that are *worse* at maximizing the AUC than the regular SVM.

Classification with the goal of AUC maximization can be seen as a special case of ordinal regression (Herbrich et al., 1999); the quadratic optimization problem that one encounters for ordinal regression correspondingly resembles Optimization Problem 3.

In AUC maximization problems, quadratically many terms contribute to the optimization function. Random or heuristic sampling has been proposed as a solution strategy (Herschtal & Raskutti, 2004). For large databases, even the execution time of the regular SVM can be unpleasant and clustering has been proposed as a strategy of reducing the number of optimization constraints and parameters (Yu et al., 2003).

7. Conclusion

We developed an AUC maximizing kernel machine that optimizes a bound on the AUC and a margin term. Starting from this optimization criterion, we derived the corresponding convex quadratic optimization problems for 1-norm and 2-norm machines that can be handled by standard QP solvers.

Our experiments with different types of kernels show that the AUC SVM generally incurs a higher AUC performance than the regular SVM. The optimal setting of the trade-off parameter is smaller than for the regular SVM. The execution time of the AUC SVM is approximately in $\mathcal{O}(n^4)$. Direct AUC maximization is feasible for small samples; for large databases, it can be approximated. The approximate k-means AUC SVM is more effective at maximizing the AUC than the SVM for linear kernels. Its execution time is quadratic in the sample size, but empirically we still observe the regular SVM to be substantially faster.

ROC analysis can be extended to multi-class problems; the volume under the multi-class ROC surface (Mossman, 1999) can be approximated by averaging multiple one-versus-one (Hand & Till, 2001) or one-versus-rest curves (Provost & Domingos, 2003). To directly maximize these criteria, the AUC SVM can be applied to the corresponding binary one-versus-one or one-versus-rest problems.

Acknowledgment

This work has been supported by the German Science Foundation DFG under grant SCHE540/10-1.

References

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, *30*, 1145–1159.

Cortes, C., & Mohri, M. (2003). AUC optimization vs. error rate. *Advances in Neural Information Processing Systems*.

Egan, J. (1975). *Signal detection theory and ROC analysis*. Academic Press.

Fawcett, T. (2001). Using rule sets to maximize ROC performance. *Proceedings of the International Conference on Data Mining*.

Ferri, C., Flach, P., & Hernandez-Orallo, J. (2002). Learning decision trees using the area under the ROC curve. *Proceedings of the International Conference on Machine Learning*.

Flach, P. A. (2003). The geometry of ROC space: Using ROC isometrics to understand machine learning metrics. *Proceedings of the International Conference on Machine Learning*.

Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (1998). An efficient boosting algorithm for combining preferences. *Proceedings of the International Conference on Machine Learning*.

Fürnkranz, J., & Flach, P. (2005). ROC'n'rule learning – towards a better understanding of covering algorithms. *Machine Learning*, *58*, in press.

Goswami, A., Jin, R., & Agrawal, G. (2004). Fast and exact out-of-core k-means clustering. *Proceedings of the IEEE International Conference on Data Mining*.

Hand, D. J., & Till, R. J. (2001). A simple generalization of the area under the ROC curve for multiple class classification problems. *Machine Learning*, *45*, 171–186.

Herbrich, R., Graepel, T., & Obermayer, K. (1999). Support vector learning for ordinal regression. *Proceedings of the International Conference on Artificial Neural Networks*.

Herschtal, A., & Raskutti, B. (2004). Optimizing the area under curve. *Proceedings of the International Conference on Machine Learning*.

Kavšek, B., Lavrač, N., & Todorovski, L. (2004). ROC analysis of example weighting in subgroup discovery. *Proceedings of the Workshop on ROC Curves and AI*.

Lusted, L. B. (1971). Signal detectability and medical decision making. *Science*, *171*, 1217–1219.

Mossman, D. (1999). Three-way ROCs. *Medical Decision Making*, *19*, 78–89.

Provost, F., & Domingos, P. (2003). Tree induction for probability-based rankings. *Machine Learning*, *52*, 199–216.

Provost, F., & Fawcett, T. (2001). Robust classification for imprecise environments. *Machine Learning*, *42*, 203–231.

Provost, F., Fawcett, T., & Kohavi, R. (1998). The case against accuracy estimation for comparing inductive algorithms. *Proceedings of the International Conference on Machine Learning*.

Rakotomamonjy, A. (2004). Optimizing AUC with SVMs. *Proceedings of the Workshop on ROC Curves and AI*.

Swets, J. A. (1996). *Signal detection theory and ROC analysis in psychological diagnostics: Collected Papers*. Lawrence Erlbaum Publishers.

Yan, L., Dodier, R., Mozer, M. C., & Wolniewicz, R. (2003). Optimizing classifier performance via the Wilcoxon-Mann-Whitney statistics. *Proceedings of the International Conference on Machine Learning*.

Yu, H., Yang, J., & Han, J. (2003). Classifying large data sets using SVMs with hierarchical clusters. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.