Structure-embedded AUC-SVM

Yunyun Wang Songcan Chen* Hui Xue

Department of Computer Science and Engineering, Nanjing University of Aeronautics & Astronautics, 210016, Nanjing, P.R. China

Abstract: AUC-SVM directly maximizes the area under the ROC curve (AUC) through minimizing its hinge loss relaxation, and the decision function is determined by those support vector *sample pairs* playing the same roles as the support vector *samples* in SVM. Such a learning paradigm generally emphasizes more on the local discriminative information just associated with these support vectors whereas hardly takes the overall view of data into account, thereby it may incur loss of the global distribution information in data favorable for classification. Moreover, due to the high computational complexity of AUC-SVM induced by the large number of training sample pairs quadratic in the number of the samples, sampling is usually adopted, whereas incurs a further loss of the distribution information in data. In order to compensate the distribution information loss and simultaneously boost the AUC-SVM performance, in this paper, we develop a novel structure-embedded AUC-SVM (SAUC-SVM for short) through embedding the global structure information in the whole data into AUC-SVM. With such an embedding, the proposed SAUC-SVM incorporates the local discriminative information and global structure information in data into a uniform formulation and consequently guarantees better generalization performance. Comparative experiments on both synthetic and real datasets confirm its effectiveness.

Key words: Area Under the ROC Curve (AUC), Support Vector Machine (SVM), Support Vector Sample Pair, Sampling, Structure Information.

1. Introduction

The area under the ROC curve (AUC) is statistically consistent and more discriminating than accuracy [6,7], and it is insensitive to imbalance prior distribution and unequal misclassification cost [3,6,7,24,25], thus recently, AUC has been widely used as an alternative

** Corresponding author: Tel: +86-25-84896481-12106, Fax: +86-25-84498069. Email: s.chen@nuaa.edu.cn (S. C. Chen)

to accuracy in evaluating the performance of classifiers [3,6,7,25]. Unfortunately, the mainstream classification algorithms are generally developed for optimizing the accuracy [1,2,8,14,18,26]. These methods may also achieve excellent AUC performance simultaneously, as the average AUC is monotonically increasing with accuracy [3]. But for imbalance distributions, classifiers with the same accuracy exhibit different AUC performances, especially when the accuracy is low [3]. As a result, it will be better to directly optimize AUC instead of accuracy, especially when the prior distribution is skew or the misclassification costs are unequal.

During the last years, several learning algorithms have been specially developed to maximize AUC, or equivalently, the Wilcoxon-Mann-Whitney statistic [1,2,8,14,18,26,27]. The WMW statistic is actually a non-differentiable objective criterion [1,8,18,27], thus direct optimization for it is not quite feasible. In terms of different approximations to it, two categories of algorithms are mainly formed [27]. One is the gradient-based approaches trying to maximize its differentiable approximations [1,18,27]. The other is the SVM-like approaches that minimize its hinge loss relaxation [2,8,14,26]. These approaches in the second category essentially maximize the relative difference of scores between samples from different classes, which is analogous to the margin in SVM [22]. Rakotomamonjy et al. [2] and Brefeld et al. [26] respectively presented the quadratic programming AUC-maximized SVMs (AUC-SVM) through minimizing the regularized hinge loss induced from AUC with the ranking constraints imposed on sample pairs, the samples in each pair come respectively from different classes. As a result, AUC-SVM actually performs on sample pairs, and naturally the quadratically-grown number of the sample pairs leads to high computation complexity, thereby some sampling method has to be adopted to mitigate such a scenario [2,26]. In addition, some linear programming variants of AUC-SVM have also been proposed [8,14] recently to partially reduce the complexity induced from the quadratic programming while retaining comparable AUC performance.

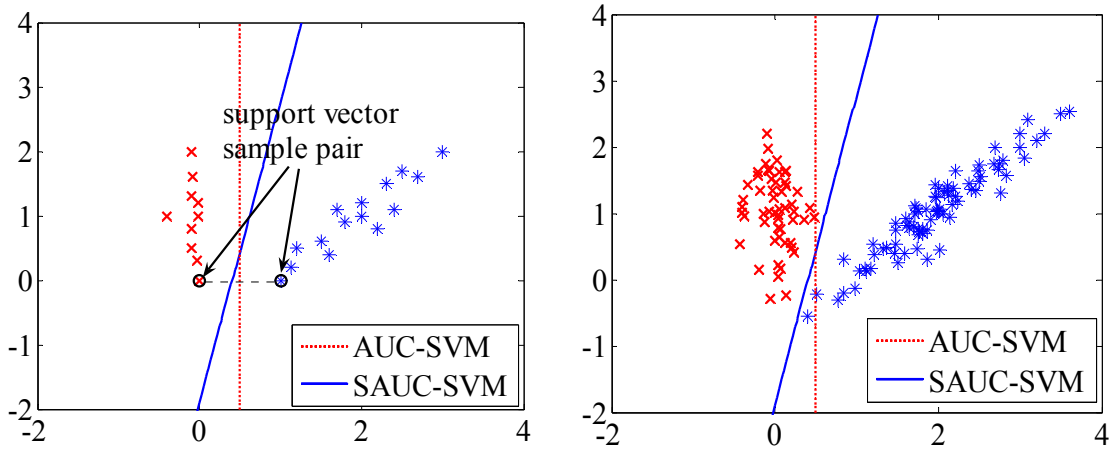
Similar to SVM [4,9,21], the decision function of AUC-SVM is determined by those support vector *sample pairs* [2,26], which are analogous to the support vector *samples* in SVM. Naturally, such two learning paradigms share a common point, i.e., usually emphasize more on the local discriminative information just associated with these sparse support vectors

whereas hardly takes the overall view of data into account [15,16,17]. More specifically, AUC-SVM only focuses on the contributions of those *sample pairs* strictly meeting the ranking constraints or badly ranked [2], thus likely misses the distribution information in the whole set of the *sample pairs*.

An illustrative example is given in Fig. 1, the two-class samples are generated independently from two Gaussian distributions and represented by '×' and '.' respectively. Through respectively optimizing the AUC-SVM and SAUC-SVM criteria (to be presented in this paper and detailed later) with the linear kernel, we can obtain their corresponding separating hyperplanes[a]. Moreover, both AUC-SVM and SAUC-SVM essentially perform on the sample pairs constructed from the given two classes, or more specifically the difference vectors of these sample pairs (these difference vectors actually give rise to an equivalent formulation of one-class SVM with the maximum margin between the difference vectors and the decision hyperplane passing through the origin [26]), thus we show the distribution of these difference vectors and the corresponding decision hyperplanes on them in Fig. 2 (actually the decision hyperplane of the equivalent one-class SVM). From Fig. 2(a), it can be seen that the dotted-line decision hyperplane derived from AUC-SVM is just determined by the support vector (*sample pair*) and perpendicular to the connected-line between the support vector and the origin, thus it hardly takes the overall view of the sample-pair set into account and is irrelevant to the other non-support-vector sample pairs. While a different solid-line decision hyperplane can be derived from SAUC-SVM through taking the distribution of the sample pair set into account, the derived hyperplane lies slantwise towards the long-axis direction of the sample-pair distribution in order to keep away from the bottom sample pairs which are more likely to be misclassified, visually, it is more reasonable and misclassifies less testing sample pairs than the vertical hyperplane from AUC-SVM does, as shown in Fig. 2(b) (the numbers of misclassified testing sample pairs are 2 and 0 respectively), therefore, it is worthy taking the distribution information of the whole sample pair set into account. Likewise, from Fig. 1, we can also see that the separating hyperplane from SAUC-SVM (solid line) is more reasonable as it achieves better testing AUC performance than that from AUC-SVM

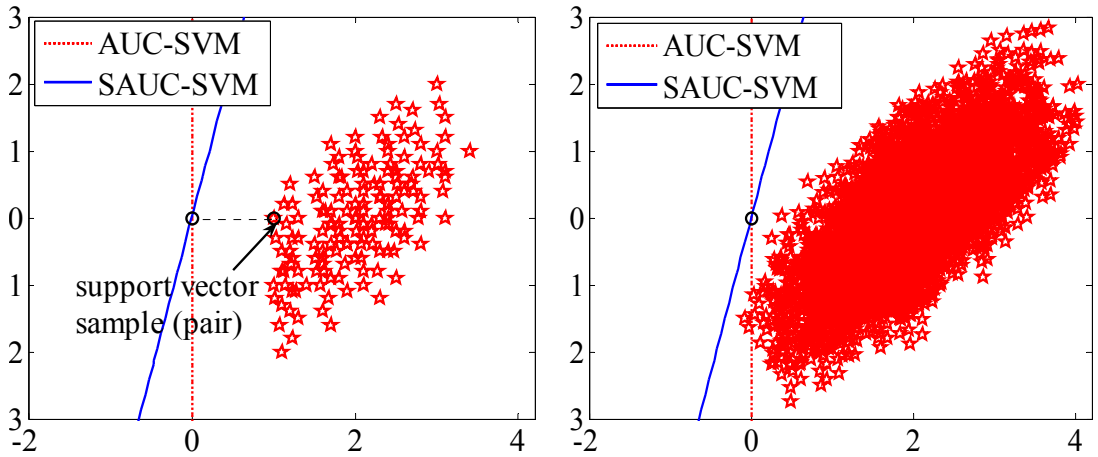
[a] The thresholds of AUC-SVM and SAUC-SVM are both determined according to the first method introduced in Section 2.

(dotted line) does, which is determined only by the support vector sample pair (the AUC values of AUC-SVM and SAUC-SVM are 0.9995 and 1 respectively). As a result, it should be reasonable to incorporate the distribution information in the sample pair set into AUC-SVM, which is specially emphasized in our proposed SAUC-SVM.



(a) (b)

Fig. 1 Binary (a) training set and (b) testing set and the corresponding separation hyperplances from AUC-SVM and SAUC-SVM respectively



(a) (b)

Fig. 2 The corresponding (a) training set and (b) testing set of the difference vectors, and the decision hyperplanes from AUC-SVM and SAUC-SVM in the *difference vector space*

Moreover, AUC-SVM actually performs on sample pairs consisted of samples from individual classes, and the number of the sample pairs is quadratic in the number of samples [2,8,14,26], naturally leading to extremely high computational complexity in learning. As a result, some sampling technique has to be adopted to reduce such complexity [2,8,26].

Undoubtedly, sampling will lead to a further loss of the distribution information contained in the whole data.

In order to compensate the information loss and simultaneously boost the performance of AUC-SVM, in this paper, we develop a novel structure-embedded AUC-SVM (SAUC-SVM for short) through embedding the structure information in the whole data into the AUC-SVM formulation, and the structure information is represented by the second-order statistics (the sample covariance matrix) here [10,11]. In fact, previous works [10,15,16,17] have confirmed that utilizing the structure information in data in learning SVM can boost its accuracy, **t**hus it can be expected that through the embedding of structure information into AUC-SVM, the proposed SAUC-SVM can guarantee better AUC performance. Comparative experiments on both toy and real datasets verify the effectiveness of our SAUC-SVM.

The rest of the paper is organized as follows. Section 2 reviews the related work about AUC and AUC-SVM. Section 3 details our proposed SAUC-SVM, including the construction of the structure information in data, formulations of both linear and kernel SAUC-SVM, probabilistic interpretations for the embedding of structure information, and finally the time complexity analysis. Section 4 shows the comparative experiments with AUC-SVM on both toy and real UCI datasets. Section 5 is the conclusion and our future work.

2. Related work

2.1 The Area under the ROC curve (AUC)

As has mentioned above, AUC is insensitive to skew class distribution and unequal classification cost, and is independent of the classification threshold [3,6,7,24,25], thus it is widely used as a performance measure as well as an optimization criteria to replace accuracy [3,6,7]. Given the positive samples $S^+ = \{x_i^+\}_{i=1}^{n^+}$ and the negative samples $S^- = \{x_j^-\}_{j=1}^{n^-}$, the AUC of a given decision function f reflects the probability that f gives a higher score for a random positive sample from S^+ than that for a random negative one from S^- [2,26], and it has been proved to be identical to the value of the Wilcoxon-Mann-Whitney statistic [1,2,8,18,26,27].

$$AUC = \frac{\sum_{i=1}^{n^+}\sum_{j=1}^{n^-} 1(f(x_i^+) > f(x_j^-))}{n^+ n^-} \tag{1}$$

where $1(f(x_i^+) > f(x_j^-))$ is the indicator function equaling 1 when $f(x_i^+) > f(x_j^-)$ and 0 otherwise. When all positive samples have higher scores than the negative ones, the AUC has the highest value 1, and 0.5 for a random assignment.

2.2 AUC maximized Support Vector Machine (AUC-SVM)

AUC-SVM directly maximizes AUC through minimizing its regularized hinge loss relaxation and the optimization problem is formulated as

$$\begin{aligned} \min_{w} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n^+}\sum_{j=1}^{n^-}\xi_{i,j} \\ s.t. \quad & w^T x_i^+ - w^T x_j^- \geq 1 - \xi_{i,j} \\ & \xi_{i,j} \geq 0, \ \ \forall\ 1 \leq i \leq n^+, 1 \leq j \leq n^- \end{aligned} \tag{2}$$

where C is a regularization parameter and $\xi_{i,j}$ is the penalty for (x_i^+, x_j^-) violating the ranking constraint, consequently, $\sum_{i=1}^{n^+}\sum_{j}^{n^-}\xi_{i,j}$ is the upper bound on the number of sample pairs violating $w^T x_i^+ - w^T x_j^- \geq 0$ [26]. The corresponding dual problem can be described as follows after introducing the kernel trick [13,21],

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n^+}\sum_{j=1}^{n^-}\alpha_{ij} - \frac{1}{2}\sum_{i,j=1}^{n^+,n^-}\sum_{u,v=1}^{n^+,n^-}\alpha_{ij}\alpha_{uv}k(x_i^+ - x_j^-, x_u^+ - x_v^-) \\ s.t. \quad & 0 \leq \alpha_{ij} \leq C, \ \ \forall\ 1 \leq i \leq n^+, 1 \leq j \leq n^- \end{aligned} \tag{3}$$

Solving the dual problem leads to the sparsity of the solution, and the decision function $f(x) = \sum_{i,j=1}^{n^+,n^-}\alpha_{ij}k(x_i^+ - x_j^-, x)$ is determined just by these support vector *sample pairs* with non-zero α_{ij} s, which strictly meet or violate the ranking constraints [2,26].

According to the above description, AUC-SVM actually performs on $\left\{(x_k^+, x_k^-) \mid \ \forall k = 1...n^+ \times n^-, x_k^+ \in S^+, x_k^- \in S^-\right\}$ with $n^+ \times n^-$ sample pairs, or more specifically, the difference vector set $\{z_k \mid z_k = x_i^+ - x_j^-,\ \forall k = 1...n^+ \times n^-,\ \forall x_i^+ \in S^+, x_j^- \in S^-\}$ constructed from the sample pairs (when mapped to the kernel space, the difference vector set is essentially $\{\phi(z_k) \mid \phi(z_k) = \phi(x_i^+) - \phi(x_j^-),\ \forall k = 1..n^+ \times n^-,\ \forall \phi(x_i^+) \in S^{+\phi}, \phi(x_j^-) \in S^{-\phi}\}$), thus some sampling method has to be adopted to reduce the high computation complexity induced by so large number of

sample pairs [2,8,26]. Likewise, the linear kernel AUC-SVM is equivalent to the unthresholded linear kernel one-class SVM trained on the set of the difference vectors [26].

Besides, due to the independence of AUC on the decision threshold, the undetermined threshold of AUC-SVM can be derived according to the specific classification task after the assignment of ranking scores to the given samples. The decision threshold can be set as

$$b = -\frac{\min f(x_i^+) + \max f(x_j^-)}{2} \tag{4}$$

where $\min f(x_i^+)$ and $\max f(x_j^-)$ denote the minimum score assigned to the positive samples and the maximum score assigned to the negative samples respectively, or we can set b to achieve the best *true positive rate-false positive rate* compromising for an equal misclassification cost, this value is obtained from the point of the ROC curve intersecting the *(0,1)-(1,0)* diagonal line [2].

3. Structure-embedded AUC-SVM

Based on the above analysis, we have clearly known that AUC-SVM emphasizes more on the local discriminative information just associated with the support vector *sample pairs*, whereas pays little attention to the global distribution of the whole set of sample pairs. Moreover, due to its high computation complexity in learning induced by the large number of training sample pairs, sampling is usually adopted [2,8,26] and thereby results in a further loss of the distribution information in the sample pair set. In order to compensate such a loss, and meanwhile motivated by the previous works of utilizing the structure information in data in learning SVM for better accuracy performance [10,15,16,17], we develop a novel structure-embedded AUC-SVM through embedding the global structure information in data (sample pair set) into AUC-SVM for boosted AUC performance. We will detail SAUC-SVM in this section, including the key construction of the structure information in data, formulations of both linear and kernel SAUC-SVM, probabilistic interpretations for such an embedding and subsequently the time complexity analysis.

3.1 Construction of the structure information in data

In order to compensate the distribution information loss and simultaneously boost the

AUC-SVM performance, we embed the global structure information in the sample pair set, or more specifically the difference vector set into AUC-SVM, which is represented by the second-order statistics (the sample covariance matrix) here [10,11]. Naturally, the size of such matrix reaches $(n^+ \times n^-) \times (n^+ \times n^-)$ and its direct computation is time consuming, just as the computation of the un-sampled kernel matrix in AUC-SVM. Fortunately, from the following equivalence proposition, we can simply use the sum of the sample covariance matrices in individual classes with far smaller size of $(n^+ + n^-) \times (n^+ + n^-)$.

Proposition 1 *Given the positive samples* $S^+ = \{x_i^+\}_{i=1}^{n^+}$ *and the negative samples* $S^- = \{x_j^-\}_{j=1}^{n^-}$, *the sample covariance matrix of their difference vector set* $\{z_k \mid z_k = x_i^+ - x_j^-,\ \forall k = 1...n^+ \times n^-,\ \forall x_i^+ \in S^+, x_j^- \in S^-\}$ *is identical to the sum of their individual sample covariance matrices.*

Proof: We first prove the equivalence in the input space, and the same equivalence in the kernel space is straightforward.

Let u_1 and u_2 denote the sample means of the two classes respectively, then the sum S_1 of their individual sample covariance matrices (Σ_1 and Σ_2) can be written as

$$\begin{aligned} S_1 &= \Sigma_1 + \Sigma_2 \\ &= \frac{1}{n^+}\sum_{i=1}^{n^+}(x_i^+ - u_1)(x_i^+ - u_1)^T + \frac{1}{n^-}\sum_{j=1}^{n^-}(x_j^- - u_2)(x_j^- - u_2)^T \\ &= \frac{1}{n^+}\sum_{i=1}^{n^+} x_i^+ x_i^{+T} - u_1 u_1^T + \frac{1}{n^-}\sum_{j=1}^{n^-} x_j^- x_j^{-T} - u_2 u_2^T \end{aligned} \tag{5}$$

And the sample covariance matrix S_2 of the difference vector set (also denoted as Σ) is

$$\begin{aligned} S_2 = \Sigma &= \frac{1}{n^+ n^-}\sum_{k=1}^{n^+ n^-}(z_k - u)(z_k - u)^T \\ &= \frac{1}{n^+}\sum_{i=1}^{n^+} x_i^+ x_i^{+T} - u_1 u_1^T + \frac{1}{n^-}\sum_{j=1}^{n^-} x_j^- x_j^{-T} - u_2 u_2^T \end{aligned} \tag{6}$$

where $u = \frac{1}{n^+ n^-}\sum_{k=1}^{n^+ n^-} z_k = \frac{1}{n^+ n^-}\sum_{i=1}^{n^+}\sum_{j=1}^{n^-}(x_i^+ - x_j^-) = u_1 - u_2$. (7)

Thus $S_1 = S_2$, i.e., we have the equivalence between S_1 and S_2 in the input space.

When generalized to the kernel space with some nonlinear mapping $\phi: R^d \to H$, the same conclusion can easily be derived from $\phi(z_k) = \phi(x_i^+) - \phi(x_j^-)$, $\forall k = 1...n^+ \times n^-$ and $u^\phi = \frac{1}{n^+ n^-}\sum_{k=1}^{n^+ n^-}\phi(z_k) = \frac{1}{n^+ n^-}\sum_{i=1}^{n^+}\sum_{j=1}^{n^-}(\phi(x_i^+) - \phi(x_j^-)) = u_1^\phi - u_2^\phi$. ■

3.2 Linear SAUC-SVM

For the purpose of clarity, we first introduce the linear version SAUC-SVM and the kernel version will be presented in the next sub-section. Inspired by our previous works [10,11], we directly embed the structure information in data into the objective function of AUC-SVM and then the linear version SAUC-SVM can be formulated as

$$\begin{aligned} \min_{w} \quad & \frac{1}{2}\| w \|^2 + C\sum_{i=1}^{n^+}\sum_{j}^{n^-} \xi_{i,j} + \frac{\lambda}{2} w^T \Sigma w \\ s.t. \quad & w^T x_i^+ - w^T x_j^- \geq 1 - \xi_{i,j} \\ & \xi_{i,j} \geq 0, \quad \forall\ 1 \leq i \leq n^+, 1 \leq j \leq n^- \end{aligned} \tag{8}$$

where Σ is the sample covariance matrix estimated from the difference vector set and $\lambda \geq 0$ is the parameter regulating the relative importance of the structure information embedded. However, we need to point out that there is an essential difference between our embedding here and [10], i.e., formally the same due to Proposition 1 but essentially different because the structure information we embedded is actually the sample covariance matrix of the difference vector set rather than the sample covariance matrices in individual classes. However, exactly due to Proposition 1, we can use the sum of sample covariance matrices in individual classes to substitute the sample covariance matrix of the difference vector set for significantly lowering the time complexity.

The new term embedded is actually a reflection of compactness within the difference vector set [10,11], as a result, SAUC-SVM maximizes the score-differences between samples from different classes and simultaneously minimizes the compactness of the difference vector set on which AUC-SVM actually performs, and it is expected to guarantee better generalization performance than AUC-SVM which concentrates only on the separation between classes.

With the standard method of Lagrange multipliers, the dual problem of (8) can be formulated as

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n^+}\sum_{j=1}^{n^-} \alpha_{ij} - \frac{1}{2}\sum_{i,j=1}^{n^+,n^-}\sum_{u,v=1}^{n^+,n^-} \alpha_{ij}\alpha_{uv}(x_i^+ - x_j^-)^T (I + \lambda\Sigma)^{-1}(x_u^+ - x_v^-) \\ s.t. \quad & 0 \leq \alpha_{ij} \leq C, \quad \forall\ 1 \leq i \leq n^+, 1 \leq j \leq n^- \end{aligned} \tag{9}$$

Consequently, SAUC-SVM is similarly performed on the sample pair

set $\left\{(x_k^+, x_k^-) \mid \ \forall k = 1...n^+ \times n^-, x_k^+ \in S^+, x_k^- \in S^-\right\}$. Now let $P^+ = \left\{x_k^+ \mid \ \forall k = 1...n^+ \times n^-, x_k^+ \in S^+\right\}$ and $P^- = \left\{x_k^- \mid \ \forall k = 1...n^+ \times n^-, x_k^- \in S^-\right\}$ from the sample pair set, respectively, then (9) can be simplified as

$$\begin{aligned} \max_{\alpha} \quad & \vec{1}^T_{n^+ \times n^-} vec(\alpha) - \frac{1}{2} vec(\alpha)^T (P^+ - P^-)^T (I + \lambda\Sigma)^{-1} (P^+ - P^-) vec(\alpha) \\ s.t. \quad & 0 \le \alpha \le C I_{n^+ \times n^-} \end{aligned} \tag{10}$$

where $\vec{1}^T_{n^+ \times n^-}$ is a $n^+ \times n^-$-dimension vector with each entry being 1, $I_{n^+ \times n^-}$ is a $n^+ \times n^-$ matrix with all elements equaling to 1 and $vec(\alpha)$ is a matrix straight operator transforming α into a corresponding column vector, in which the sequence of elements accords with the sequences of the sample pairs in P^+ and P^-, finally the scoring function can be obtained from

$$f(x) = vec(\alpha)^T (P^+ - P^-)^T (I + \lambda\Sigma)^{-1} x \tag{11}$$

It is clear that the optimization problem of SAUC-SVM is still solved as a QP problem, and due to the large number of training sample pairs quadratic in the number of the samples, some sampling method has to be adopted to reduce the resulting high computation complexity. Naturally, sampling will incur loss of information in data and thus bring a decline in the AUC performance, just as in AUC-SVM. However, it is the embedding of the structure information (the 2nd statistics here) summarized from the whole data into AUC-SVM that the newly-generated SAUC-SVM can compensate the loss incurred by sampling as has mentioned. The sampling method we used in this paper follows that in [2]: instead of using all $n^+ \times n^-$ sample pairs constructed from individual classes, only those sample pairs consisted of the samples and their *k* nearest neighbors in the opposite class are selected for training, these selected sample pairs are more likely to be support vector *sample pairs*, which are much analogous to the would-be support vectors of boundary samples in SVM [12].

3.3 Kernel SAUC-SVM

For many linearly inseparable classification cases in the real world, a nonlinear mapping of data from the input space to a high (even infinite) dimension kernel space would make them linearly separable with high probability in the kernel space. Then through the implementation

of the original algorithm in such space, a corresponding nonlinear decision function in the original input space can be obtained with better classification. However, due to high dimensionality of the kernel space, the nonlinear mapping can not be formulated explicitly, fortunately, if all calculations between sample pairs in the original algorithm can be expressed in the form of inner products, these inner products can be replaced by a reproducing kernel to avoid the direct reference to the nonlinear mapping, which is the so-called kernel trick [13,21]. In this subsection, we will apply the kernel trick to SAUC-SVM to solve the linear inseparable cases in the real world.

Let us first define an implicit nonlinear kernel mapping $\phi: R^d \to H$, where H is a Reproducing Kernel Hilbert Space (RKHS), the kernel version SAUC-SVM can be formulated as

$$\begin{aligned} \min_{w} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n^+}\sum_{j}^{n^-}\xi_{i,j} + \frac{\lambda}{2}w^T\Sigma^{\phi}w \\ s.t. \quad & w^T\phi(x_i^+) - w^T\phi(x_j^-) \geq 1 - \xi_{i,j} \\ & \xi_{i,j} \geq 0, \quad \forall\ 1 \leq i \leq n^+, 1 \leq j \leq n^- \end{aligned} \tag{12}$$

where Σ^{ϕ} is the sample covariance matrix of the difference vector set in the kernel space.

Accordingly, the corresponding dual formulation is

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n^+}\sum_{j=1}^{n^-}\alpha_{ij} - \frac{1}{2}\sum_{i,j=1}^{n^+,n^-}\sum_{u,v=1}^{n^+,n^-}\alpha_{ij}\alpha_{uv}(\phi(x_i^+) - \phi(x_j^-))^T(I + \lambda\Sigma^{\phi})^{-1}(\phi(x_u^+) - \phi(x_v^-)) \\ s.t. \quad & 0 \leq \alpha_{ij} \leq C, \quad \forall\ 1 \leq i \leq n^+, 1 \leq j \leq n^- \end{aligned} \tag{13}$$

which can further be rewritten as

$$\begin{aligned} \max_{\alpha} \quad & \vec{1}_{n^+ \times n^-}^T vec(\alpha) - \frac{1}{2}vec(\alpha)^T(P^{+\phi} - P^{-\phi})^T(I + \lambda\Sigma^{\phi})^{-1}(P^{+\phi} - P^{-\phi})vec(\alpha) \\ s.t. \quad & 0 \leq \alpha \leq CI_{n^+ \times n^-} \end{aligned} \tag{14}$$

where $P^{+\phi}$ and $P^{-\phi}$ are the corresponding notations of P^+ and P^- in the kernel space.

Further, all calculations in (14) can actually be expressed in the form of inner products. In order to make clear of it, let us first convert the sample covariance matrix of the difference vector set in the kernel space to the following form,

$$
\begin{aligned}
\Sigma^{\phi} &= \frac{1}{n^{+}n^{-}} \sum_{k=1}^{n^{+}n^{-}} (\phi(z_k) - u^{\phi})(\phi(z_k) - u^{\phi})^T \\
&= S^{+\phi}(\frac{1}{n^{+}} I_{n^{+}} - \vec{1}_{n^{+}} \vec{1}_{n^{+}}{}^T) S^{+\phi T} + S^{-\phi}(\frac{1}{n^{-}} I_{n^{-}} - \vec{1}_{n^{-}} \vec{1}_{n^{-}}{}^T) S^{-\phi T} \\
&= [S^{+\phi} \quad S^{-\phi}] \begin{bmatrix} \frac{1}{n^{+}} I_{n^{+}} - \vec{1}_{n^{+}} \vec{1}_{n^{+}}{}^T & 0_{n^{+}} \\ 0_{n^{-}} & \frac{1}{n^{-}} I_{n^{-}} - \vec{1}_{n^{-}} \vec{1}_{n^{-}}{}^T \end{bmatrix} \begin{bmatrix} S^{+\phi T} \\ S^{-\phi T} \end{bmatrix} \\
&= S^{\phi} R S^{\phi T}
\end{aligned}
\tag{15}
$$

where $I_n^{+}(I_n^{-})$ is a $n^{+} \times n^{+}(n^{-} \times n^{-})$ identity matrix and $\vec{1}_{n^{+}}$ ($\vec{1}_{n^{-}}$) is a $n^{+}(n^{-})$-dimension vector with all elements being $1/n^{+}(1/n^{-})$. $S^{\phi} = [S^{+\phi} \quad S^{-\phi}]$ is the matrix consisted of all training samples in the kernel space and *R* denotes the symmetry matrix between S^{ϕ} and $S^{\phi T}$ in (15). At the same time, by the following Woodbury's formula [20]

$$
(A + UBV)^{-1} = A^{-1} - A^{-1}UB(B + BVA^{-1}UB)^{-1}BVA^{-1} \tag{16}
$$

we have

$$
(I + \lambda \Sigma^{\phi})^{-1} = I - \lambda S^{\phi} R(R + \lambda R S^{\phi T} S^{\phi} R)^{-1} R S^{\phi T} \tag{17}
$$

Finally the dual formulation of SAUC-SVM in the kernel space can be described as

$$
\begin{aligned}
\max_{\alpha} \ & \vec{1}_{n^{+} \times n^{-}}^{T} vec(\alpha) - \frac{1}{2} vec(\alpha)^T [(P^{+\phi} - P^{-\phi})^T (P^{+\phi} - P^{-\phi}) - \\
& \lambda (P^{+\phi} - P^{-\phi})^T S^{\phi} R(R + \lambda R S^{\phi T} S^{\phi} R)^{-1} R S^{\phi T} (P^{+\phi} - P^{-\phi})] vec(\alpha) \\
s.t. \quad & 0 \le \alpha \le C I_{n^{+} \times n^{-}}
\end{aligned}
\tag{18}
$$

Let $K = (P^{+\phi} - P^{-\phi})^T (P^{+\phi} - P^{-\phi})$, $K^{+} = P^{+\phi T} S^{\phi}$, $K^{-} = P^{-\phi T} S^{\phi}$ and $\tilde{K} = S^{\phi T} S^{\phi}$, each entry in them is exactly an inner-product between sample pairs in the kernel space and can be replaced by a reproducing kernel, then *K* and $\tilde{K}$ are the kernel matrices in AUC-SVM and SVM respectively. Finally, (18) can be re-written as

$$
\begin{aligned}
\max_{\alpha} \ & \vec{1}_{n^{+} \times n^{-}}^{T} vec(\alpha) - \frac{1}{2} vec(\alpha)^T [K - \lambda (K^{+} - K^{-}) R(R + \lambda R \tilde{K} R)^{-1} R (K^{+} - K^{-})^T] vec(\alpha) \\
s.t. \quad & 0 \le \alpha \le C I_{n^{+} \times n^{-}}
\end{aligned}
\tag{19}
$$

and the scoring function of SAUC-SVM can be formulated as

$$
f(x) = vec(\alpha)^T [(K_x^{+} - K_x^{-}) - (K^{+} - K^{-}) R(R + \lambda R \tilde{K} R)^{-1} R \tilde{K}_x] \tag{20}
$$

where K_x^+, K_x^- and $\tilde{K}_x$ are the respective columns in K^+, K^- and $\tilde{K}$ corresponding to *x*.

3.4 Probability interpretation

In this subsection, we will provide two probabilistic interpretations for the reasonability of embedding the structure information into AUC-SVM, from the specific Gaussian distribution cases to the general cases without any assumption for the data distributions.

Proposition 2 *Given the positive samples* $S^+ = \{x_i^+\}_{i=1}^{n^+}$ *and the negative samples* $S^- = \{x_j^-\}_{j=1}^{n^-}$ *with individual means of* u^+ *and* u^-*, and individual covariance matrices of* Σ^+ *and* Σ^-*, AUC-SVM concentrates on the maximization of the score-difference between samples from different classes, actually leading to the maximization of* $w^T(u_1 - u_2)$*. While SAUC-SVM also takes the structure information in the difference vector set into account, and thus maximizes* $w^T(u_1 - u_2)$ *and minimizes* $w^T(\Sigma_1 + \Sigma_2)w$ *simultaneously.*

Proof: From the same ranking constraints of both AUC-SVM in (2) and SAUC-SVM in (8), we have

$$\xi_{ij} \geq 1 - w^T(x_i^+ - x_j^-), \quad \forall\ 1 \leq i \leq n^+, 1 \leq j \leq n^- \tag{21}$$

then

$$\begin{aligned} \sum_{i=1}^{n^+}\sum_{j=1}^{n^-}\xi_{ij} &\geq n^+n^- - \sum_{i=1}^{n^+}\sum_{j=1}^{n^-} w^T(x_i^+ - x_j^-) \\ &= n^+n^- - n^+n^- w^T(u_1 - u_2) \end{aligned} \tag{22}$$

Therefore, minimizing $\sum_{i=1}^{n^+}\sum_{j}^{n^-}\xi_{i,j}$ in the objective functions of both AUC-SVM and SAUC-SVM leads to the maximization of $w^T(u_1 - u_2)$. Further, SAUC-SVM embeds the structure information in the difference vector set into AUC-SVM, and thus minimizes $w^T(\Sigma_1 + \Sigma_2)w$ at the same time. ■

Below we will give a probabilistic interpretation for the reasonability of embedding the structure information in data into AUC-SVM, specifically the minimization of $w^T(\Sigma_1 + \Sigma_2)w$ simply under the assumption of Gaussian distributions.

Suppose two Gaussian distributions $X^+ \sim N(u^+, \Sigma^+)$ and $X^- \sim N(u^-, \Sigma^-)$ respectively

corresponding to two classes. In order to maximize AUC, we try to find a w to maximize the probability $\Pr(w^T X^+ - w^T X^- > 0)$, by property of the Gaussian distribution, the set of the difference vectors $X^+ - X^-$ also follows a Gaussian distribution with mean u and covariance matrix Σ, where $u = u^+ - u^-$ and $\Sigma = \Sigma^+ + \Sigma^-$, as a result, we have

$$\begin{aligned} &\Pr(w^T X^+ - w^T X^- > 0) \\ &= \Psi\left(\frac{w^T (u^+ - u^-)}{\sqrt{w^T (\Sigma^+ + \Sigma^-) w}}\right) \end{aligned} \tag{23}$$

where $\Psi(\cdot)$ is the cumulative distribution function of the standard normal distribution. Due to its monotonicity, we can equivalently maximize $\frac{w^T (u^+ - u^-)}{\sqrt{w^T (\Sigma^+ + \Sigma^-) w}}$, i.e., we should maximize $w^T (u^+ - u^-)$ and minimize $w^T (\Sigma^+ + \Sigma^-) w$ simultaneously. In this way, it is reasonable to embed the structure information in data into the original AUC-SVM.

The above analysis is for the specific Gaussian distribution cases. Below we will provide a more practical interpretation without any assumption for the data distributions.

Suppose the binary samples X^+ and X^- are generated independently from two distributions with individual means of u^+ and u^-, and individual covariance matrices of Σ^+ and Σ^-. Then the mean and covariance matrix of their difference vectors $Z = X^+ - X^-$ are $u = u^+ - u^-$ and $\Sigma = \Sigma^+ + \Sigma^-$ respectively. The classification objective here is to find a w to maximize $\Pr(w^T X^+ > w^T X^-)$, i.e. $\Pr(w^T Z > 0)$ with $E(w^T Z) = w^T u$ and $Cov(w^T Z) = w^T \Sigma w$. From the Chebyshev bound [23],

$$\begin{aligned} \Pr(w^T Z > 0) &\geq \Theta\left(\frac{w^T u}{\sqrt{w^T \Sigma w}}\right) \\ &= \Theta\left(\frac{w^T (u^+ - u^-)}{\sqrt{w^T (\Sigma^+ + \Sigma^-) w}}\right) \end{aligned} \tag{24}$$

where

$$\Theta(x) = \frac{\max(x, 0)^2}{1 + \max(x, 0)^2}. \tag{25}$$

Clearly, directly maximizing the probability $\Pr(w^T Z > 0)$ is hard without any assumption

for the data distributions, instead, we can maximize its lower bound $\Theta(\frac{w^T(u^+-u^-)}{\sqrt{w^T(\Sigma^+ + \Sigma^-)w}})$ in (24). Further, since $\Theta(\cdot)$ is an increasing function, we can equivalently maximize $\frac{w^T(u^+-u^-)}{\sqrt{w^T(\Sigma^+ + \Sigma^-)w}}$, i.e., we should maximize $w^T(u^+-u^-)$ and minimize $w^T(\Sigma^+ + \Sigma^-)w$ simultaneously.

As a result, it is worthy embedding the structure information in the whole data into AUC-SVM, and naturally through such embedding, it can be expected to obtain better AUC performance from the proposed SAUC-SVM.

3.5 Time complexity analysis

Given the data $X \in R^{d\times n}$, where n is the number of samples with d dimension features, AUC-SVM actually performs on the difference vectors consisted of samples from different classes and the number of those difference vectors is quadratic in the number of samples, then the training time of AUC-SVM is $O(n^6)$ (the time for training SVM is $O(n^3)$ [4]). After the embedding of the structure information in data, the proposed SAUC-SVM introduces $(I+\lambda\Sigma)^{-1}$, actually converted to $I-\lambda SR(R+\lambda RS^TSR)^{-1}RS^T$ as shown in (17) according to the Woodbury's formula [20], then the additional computation needed in SAUC-SVM is $\lambda(K^+-K^-)R(R+\lambda R\tilde{K}R)^{-1}R(K^+-K^-)^T$ in (19) and it costs $O(n^4)$. Thereby the training time for SAUC-SVM is $O(n^4+n^6)$, i.e., $O(n^6)$ and comparable to that of AUC-SVM.

In order to reduce such a high computation complexity, sampling is usually adopted [2,8,26]. Suppose n_s sample pairs to be selected in sampling, then the training time of AUC-SVM becomes $O(n_s^3)$, and the additional computation time needed in SAUC-SVM is $O(n^3+n_s^2\times n)$, finally the training time of SAUC-SVM is $O(n^3+n_s^2\times n+n_s^3)$. To achieve satisfactory AUC performance in both AUC-SVM and SAUC-SVM, the number of sample pairs selected in sampling is usually larger than the number of samples, i.e. $n_s \geq n$ [2,8,26], then the training time of SAUC-SVM is $O(n_s^3)$, still comparable to that of AUC-SVM.

In conclusion, after the embedding of the structure information in data, the proposed SAUC-SVM can retain comparable efficiency to the original AUC-SVM.

4. Experiment

To evaluate the performance of the proposed SAUC-SVM, we perform experiments on both toy and real datasets. In the toy problem, we use a linearly inseparable XOR dataset and compare SAUC-SVM with both AUC-SVM and SVM, all with the Gaussian kernel. In the real problem, we compare these three algorithms on 9 UCI datasets (actually 18 binary datasets[b] generated) [5] using both the linear and Gaussian kernels. We adopt the sampling method presented in [2] to reduce the high computation complexity in both AUC-SVM and SAUC-SVM. We select the regularization parameters C, λ from the set $\{0.001,0.01,0.1,1,10,100,1000\}$ and the width parameter σ of the Gaussian kernel from $\{2^{-10},2^{-9},\ldots,2^{9},2^{10}\}$ respectively through the 5-fold cross-validation. We implement all algorithms in MATLAB 7.3 (R2006b) and carry out the experiments on an Intel (R) Pentium (R) dual-core 1.60GHz processor with memory 1GB, for the optimization problems, we resort to the Mosek toolbox [19] implemented in C.

4.1 Toy problem

The two-dimension XOR dataset is randomly generated from four Gaussian distributions, either two diagonal distributions belong to the same class and the attributes of the dataset are described in Table 1. We randomly select 1/5 samples in each distribution for training and the rest for testing, as shown in Fig. 2, samples in individual classes are represented by '.' and 'o' respectively.

Table 1 The attributes of the toy XOR dataset

	Mean	Covariance	Sample number
Class I	[0.8,0]	$\begin{bmatrix} 0.15 & 0.1 \\ 0.1 & 0.15 \end{bmatrix}$	100
	[-0.8,0]		100
Class II	[0, 0.8]	$\begin{bmatrix} 0.15 & -0.1 \\ -0.1 & 0.15 \end{bmatrix}$	200
	[0, -0.8]		200

[b] Multi-class datasets 'Glass' and 'Tae' have been used here and the One vs. All strategy is adopted to generate 18 two-class datasets from the original 9 UCI datasets.

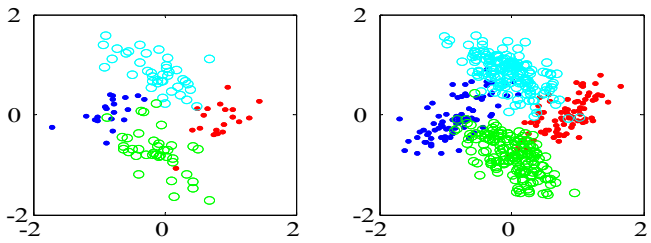


Fig. 2 The distribution of the toy (a) training dataset and (b) testing dataset

4.1.1 AUC performance comparison

We first compare the performances of SAUC-SVM, AUC-SVM and SVM with respect to different values of the nearest-neighbor number during sampling (k). The results are revealed in Fig. 3, in which the x-axis gives the values of k with the corresponding sampling ratios in the parentheses (k=INF corresponds to no sampling, i.e., using all sample pairs), and the y-axis gives the AUC performances. Then from the figure, we can make the following observations,

- When k is small and the corresponding sampling ratio is low, the performance of AUC-SVM is far worse than that of SVM, and such difference decreases with the increase of k, while all sample pairs are used, AUC-SVM performs better than SVM (due to the skew distribution of such toy dataset), then we can say that sampling does incur information loss and performance decline in AUC-SVM. However, it is clearly that the performances of SAUC-SVM with respect to different values of k are consistently better than those of both SVM and AUC-SVM. Thus through embedding the structure information in data, SAUC-SVM can really compensate the information loss incurred by sampling and naturally guarantees better AUC performance.
- When all sample pairs are used, SAUC-SVM still performs better than AUC-SVM, though with lower improvement compared to the sampling cases, implying that SAUC-SVM can boost the performance of AUC-SVM concentrating on only the local discriminative information of support vectors.
- The performance of SAUC-SVM with k=5 (sampling ratio 4.53%) here is already better than that of AUC-SVM trained on all sample pairs. As a result, we can obtain satisfactory performance with SAUC-SVM on a relatively small number of selected sample pairs with far lower computation complexity.

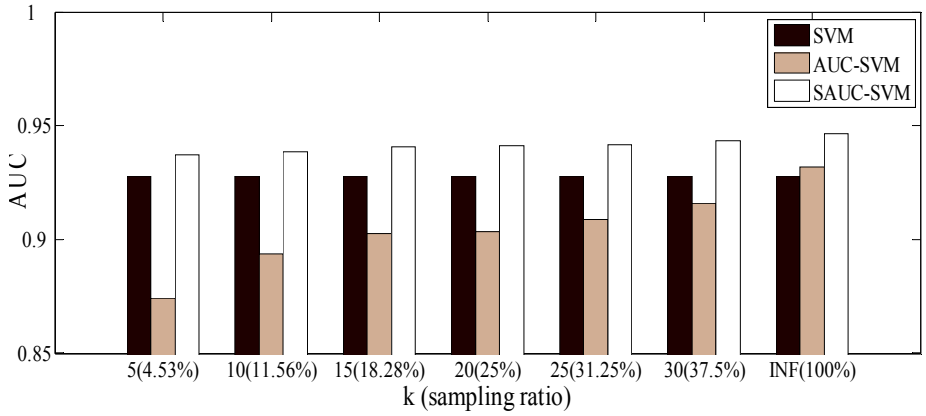


Fig. 3 The AUC performances of SVM, AUC-SVM and SAUC-SVM corresponding to different values of *k* with sampling ratios in the parentheses

4.1.2 Computation time comparison

Due to the embedding of the structure information in data, SAUC-SVM introduces additional computation cost. Thereby we provide the training times of SAUC-SVM, AUC-SVM and SVM with respect to different values of *k* in Table 2, as well as the number of training samples or sample pairs selected in those algorithms, and the iteration numbers in their individual optimizations.

Table 2 The times (s) for training SVM, AUC-SVM and SAUC-SVM with respect to different *k*, and the corresponding numbers of training samples (pairs) and iterations in optimization

		5	10	15	20	25	30	INF
SVM	Training time	0.0981	0.0981	0.0981	0.0981	0.0981	0.0981	0.0981
	Samp./Iter. No.	120/11	120/11	120/11	120/11	120/11	120/11	120/11
AUC -SVM	Training time	0.125	**0.7656**	1.4844	**3.3594**	**5.625**	**8**	**89.06**
	Samp./Iter. No.	145/**11**	370/**13**	585/**15**	800/**16**	1000/**18**	1200/**16**	3200/**24**
SAUC -SVM	Training time	0.1488	**0.620**	1.6406	**2.9062**	**5.1875**	**7.3594**	**63.5**
	Samp./Iter. No.	145/**7**	370/**10**	585/**12**	800/**8**	1000/**13**	1200/**12**	3200/**20**

Table 2 shows that the training times of both AUC-SVM and SAUC-SVM are longer than that of SVM as the numbers of training sample pairs selected here are all larger than the number of samples. However, it is interesting that with additional computation, SAUC-SVM can retain comparable or even shorter training time than AUC-SVM, which is actually due to the different iteration numbers in their individual optimizations, as shown in Table 2, for different values of *k*, SAUC-SVM needs consistently fewer iteration numbers in optimization than AUC-SVM does. As a result, after the embedding of structure information, SAUC-SVM can still retain comparable efficiency to AUC-SVM.

4.2 Real problem

In order to further investigate the effectiveness of SAUC-SVM, we compare SAUC-SVM, AUC-SVM and SVM on 18 real datasets with both the linear and Gaussian kernels. We randomly divide each dataset into two non-overlapping subsets with nearly equal number of samples, one for training and the other for testing, this process for each algorithm is repeated 10 times and their average results are reported.

4.2.1 AUC performance comparison

First, the number of the nearest neighbors selected during sampling is set to 10 following [2] and the AUC means and variances (in parentheses) on these real datasets are reported in Table 3. For each kernel, a bold value in each row indicates a better AUC performance between SAUC-SVM and AUC-SVM and an underlined value indicates a better AUC performance between SAUC-SVM and SVM, the unmarked values in each row indicates that the corresponding two algorithms have comparable performances (actually the difference of individual AUC values is less than one percent) on the dataset. The last row gives the average AUC performances of those three algorithms respectively on all the datasets used here. From Table 3, clearly,

- When the linear kernel is used, SAUC-SVM outperforms AUC-SVM on 12 datasets with the maximum improvement of 7.09% on Tae3, and retains comparable performances on the remaining 6 datasets. When the Gaussian kernel is used, SAUC-SVM performs better than AUC-SVM on 15 datasets with the maximum improvement of 4.33% on Spectf, and comparable to AUC-SVM on the remaining 3 datasets. In terms of the average AUC performances on all datasets used here, SAUC-SVM outperforms AUC-SVM by 2.45% with the linear kernel and 2.1% with the Gaussian kernel. Thus we can conclude that through the embedding of the structure information in data, SAUC-SVM can boost the AUC performance of AUC-SVM.
- It has been reported that AUC-SVM with sampling usually does not give rise to better AUC performance than SVM [2], which is also reflected in our experiments. However, our proposed SAUC-SVM still outperforms SVM on most datasets used here, and in terms of the average AUC performances on all datasets used, SAUC-SVM outperforms SVM by 1.24% with the linear kernel and 2.07% with the Gaussian kernel, which

confirms again that the embedding of structure information in data is helpful for boosting the AUC performance of AUC-SVM.

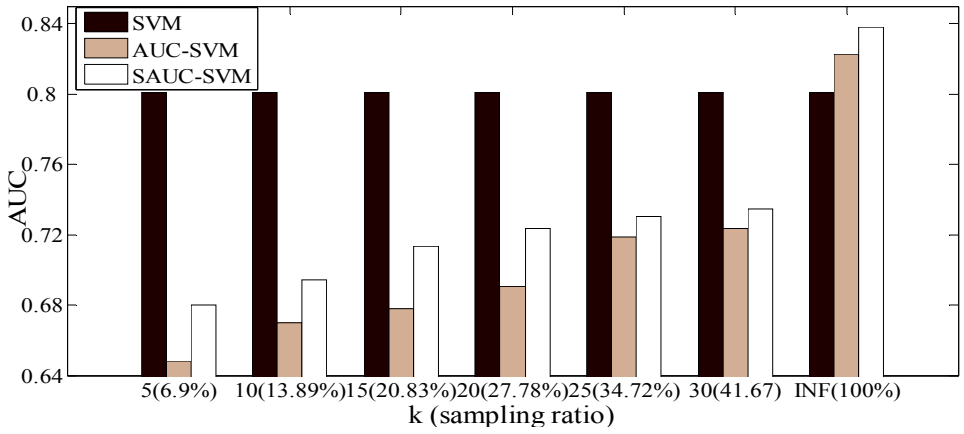
- At the same time, we find some cases in which SAUC-SVM exhibits relatively inferior performances to SVM (e.g. arrhythmia, glass1, glass2), which is mainly due to the sampling method we used here (including empirically setting k to 10), when all sample pairs are used shown in Fig. 4, both AUC-SVM and SAUC-SVM outperform SVM. As a result, the embedding of structure information can compensate the information loss incurred by sampling rather than overcome it, and some better techniques for reducing the computation complexity of AUC-SVM are still needed.

Table 3 The average results of AUC and variance (in parentheses) on 18 real datasets with both the linear and Gaussian kernels

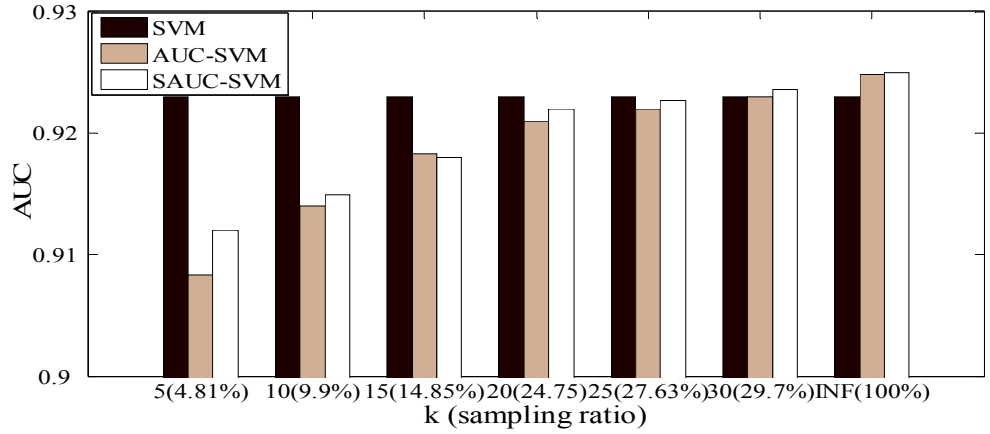
Data set	Linear kernel			Gaussian kernel		
	SVM	AUC-SVM	SAUC-SVM	SVM	AUC-SVM	SAUC-SVM
Automobile	0.901(0.0023)	0.9023(0.0013)	**0.9149(0.0011)**	0.9245(0.0011)	0.9311(0.0009)	0.9388(0.0005)
Arrhythmia	0.7018(0.001)	0.6533(0.0014)	0.6521(0.0013)	0.8031(0.0008)	0.7616(0.0011)	**0.7823(0.001)**
Bupa	0.6602(0.0102)	0.6547(0.0008)	**0.6827(0.0005)**	0.7637(0.0003)	0.8229(0.0001)	0.8293(0.0001)
Diabetes	0.7584(0.0005)	0.7892(0.0017)	**0.8114(0.0018)**	0.8123(0.0016)	0.8075(0.0013)	**0.8361(0.001)**
Glass1	0.8012(0.0048)	0.6698(0.0021)	**0.6941(0.0032)**	0.7898(0.0023)	0.7235(0.006)	**0.7581(0.0012)**
Glass2	0.6352(0.0024)	0.5869(0.0043)	**0.6057(0.0033)**	0.8408(0.001)	0.7983(0.004)	**0.826(0.0025)**
Glass3	0.507(0.01)	0.4959(0.0107)	**0.5243(0.0061)**	0.5928(0.0148)	0.6601(0.0128)	**0.6893(0.0102)**
Glass4	0.923(0.0015)	0.914(0.0019)	0.9143(0.0018)	0.9128(0.0058)	0.9267(0.0044)	**0.9588(0.0036)**
Glass5	0.9885(0.0001)	0.9395(0.0019)	**0.9740(0.0001)**	0.9765(0.0005)	0.9473(0.0016)	**0.969(0.0013)**
Glass6	0.9293(0.0015)	0.9456(0.0014)	0.9368(0.0015)	0.9584(0.0004)	0.979(0.0001)	**0.9918(0.0001)**
Hepatitis	0.6648(0.0079)	0.7066(0.005)	**0.7463(0.004)**	0.65(0.0012)	0.6913(0.0011)	**0.7056(0.0005)**
Ionosphere	0.8385(0.0015)	0.8336(0.0007)	0.8397(0.0007)	0.976(0.0001)	0.9583(0.0001)	**0.9689(0.0001)**
Import	0.9033(0.0014)	0.8792(0.0016)	**0.9143(0.0012)**	0.9362(0.0006)	0.9532(0.0002)	**0.9722(0.0001)**
Spectf	0.6909(0.0027)	0.7039(0.0026)	0.7114(0.0022)	0.7611(0.0098)	0.7798(0.0028)	**0.8231(0.0017)**
Tae1	0.6192(0.005)	0.6725(0.003)	**0.7372(0.0026)**	0.6096(0.0188)	0.5938(0.0068)	**0.6247(0.0068)**
Tae2	0.635(0.0045)	0.6156(0.0011)	**0.6704(0.0007)**	0.6581(0.0055)	0.6469(0.0024)	**0.6675(0.0043)**
Tae3	0.6545(0.0057)	0.5959(0.0019)	**0.6668(0.002)**	0.6199(0.0044)	0.6223(0.0114)	**0.6333(0.0074)**
wdbc	0.9472(0.0023)	0.9834(0.0005)	0.9868(0.0004)	0.9774(0.0001)	0.9723(0.0002)	0.9795(0.0001)
Average	0.7644	0.7523	**0.7768**	0.8091	0.8098	**0.8308**

We also show the performance comparisons on 4 selected datasets when different values of *k* are used. The results are revealed in Fig. 4, from which we can see that

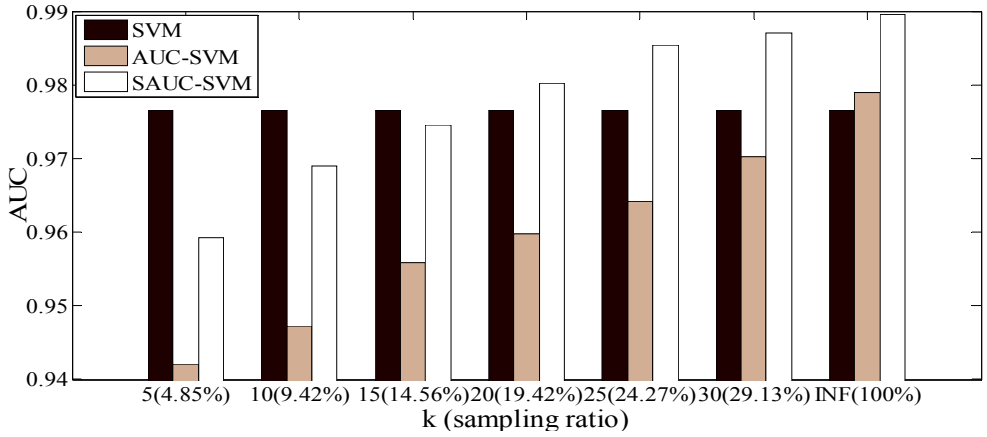
- With different values of *k*, SAUC-SVM performs consistently better than AUC-SVM, and the smaller *k* is, the larger the performance improvement from SAUC-SVM is, as there is more information loss incurred by sampling in such cases, implying that SAUC-SVM can compensate the information loss incurred by sampling.
- When all sample pairs are used, SAUC-SVM still outperforms AUC-SVM, confirming again that SAUC-SVM can boost the AUC performance of AUC-SVM concentrating on only the local discriminative information of support vectors.
- The performances of both AUC-SVM and SAUC-SVM increase with the increase of *k*, and when *k*=*INF* (all sample pairs used), both AUC-SVM and SAUC-SVM outperform SVM, as they are designed to directly optimize AUC and the class distributions are not strictly balanced here.



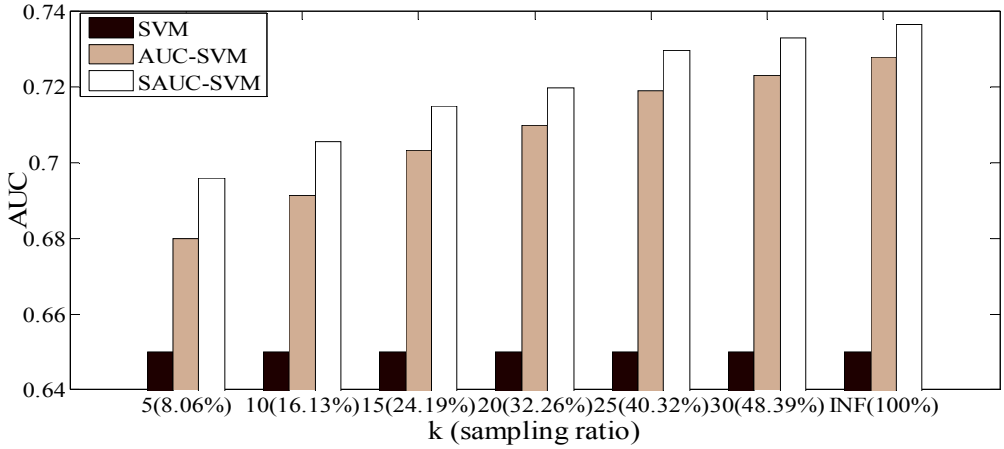
(a)



(b)



(c)



(d)

Fig. 4 Performance comparisons on 4 selected datasets when different values of k are used,

(a) glass1 with the linear kernel (b) glass4 with the linear kernel

(c) glass5 with the Gaussian kernel (d) hepatitis with the Gaussian kernel

4.2.2 Analysis on the regularization parameter λ

We also show the performances of SAUC-SVM with respect to different values of λ from $\{0,10^{-5},10^{-4},\ldots 10^{4},10^{5}\}$ in Fig. 7, with the parameters k, C and σ set to 10, 1, and 256 respectively. From this figure, different optimal values of λ can be selected for different datasets, while the ranges under consideration are all [0.001, 1000]. Furthermore, when λ=0, SAUC-SVM degenerates to the original AUC-SVM concentrating on the separation between classes, and when λ is large enough, SAUC-SVM is dominated by the new term we incorporated reflecting the within-class compactness, then we can find in Fig. 7 that the optimal AUC performances are obtained through simultaneously considering the between-class margin and the within-class compactness, just as analyzed in subsection 3.4.

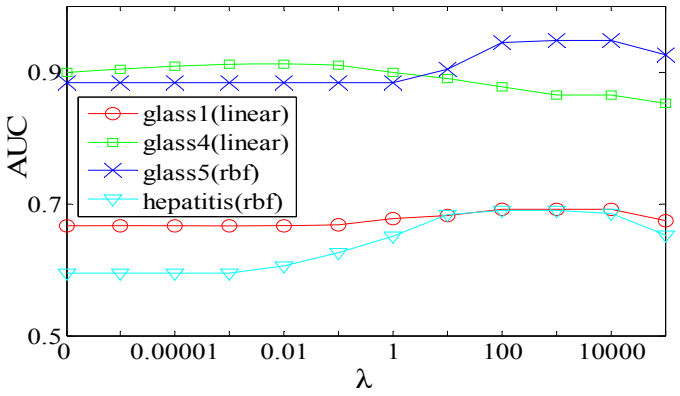


Fig. 5 The performances of SAUC-SVM with respect to different values of λ from $[0,10^{-5},10^{-4},\ldots 10^{4},10^{5}]$ on the 4 selected datasets

4.2.3 Computation time comparison

Table 4 reports the training time of these three algorithms with k=10, and the numbers of training samples or sample pairs selected in individual algorithms in the parentheses.

Table 4 The time (s) for training SVM, AUC-SVM and SAUC-SVM with k=10, as well as the individual number of training samples or sample pairs selected

Data set	Linear kernel			RBF kernel		
	SVM	AUC-SVM	SAUC-SVM	SVM	AUC-SVM	SAUC-SVM
Automobile	.0281(80)	.2359(317)	.2531 (317)	.0219(80)	.3625(317)	.3969 (317)
Arrhythmia	.1844(211)	1.9516(1118)	2.3812 (1118)	.2594(211)	1.812(1121)	2.385 (1121)
Bupa	.0734(173)	1.2016(723)	1.3484 (723)	.0641(173)	1.258(723)	1.422 (723)
Diabetes	.3609(384)	1.532(1305)	1.87 (1305)	.4031(384)	1.736(1305)	2.109 (1305)
Glass1	.0281(107)	.0813(350)	.1046 (350)	.0279(107)	.0809(323)	.0987 (350)
Glass2	.0313(107)	.097(372)	.1009 (372)	.0375(107)	.0919(372)	.1069 (372)
Glass3	.0359(108)	.0197(90)	.0357 (90)	.0375(108)	.0137(90)	.0369 (90)
Glass4	.0328(108)	.0141(70)	.0308 (70)	.0344(108)	.0169(70)	.0381 (70)
Glass5	.0391(108)	.0103(50)	.0354 (50)	.0359(108)	.0128(50)	.0309 (50)
Glass6	.0297(108)	.0231(150)	.0338 (150)	.0359(108)	.035(146)	.0482 (150)
Hepatitis	.0234(78)	.0272(160)	.0375 (160)	.0234(78)	.0253(160)	.0359 (160)
Ionosphere	.0766(176)	0.6156(953)	.6984 (953)	.0766(176)	.6156(953)	.7069 (953)
Import	.0281(80)	.1081(377)	.1187 (377)	.0203(80)	.1078(377)	.1003 (377)
Spectf	.0484(134)	.0534(280)	.0739 (280)	.0484(134)	.0531(244)	.059 (262)
Tae1	.0187(76)	.0725(250)	.0793 (250)	.0234(76)	.0898(250)	.087 (250)

Tae2	.0219(76)	.0825(250)	.105 (250)	.0156(76)	.0922(250)	.0884 (240)
Tae3	.0187(76)	.0737(260)	.0841 (260)	.0328(76)	.0971(260)	.0934(260)
wdbc	.3125(285)	.2562(402)	.3225 (402)	.225(285)	.2156(402)	.3612 (402)

From Table 4, for satisfactory AUC performances, the numbers of sample pairs selected in both AUC-SVM and SAUC-SVM should be larger than the corresponding numbers of the given samples on most datasets, naturally their training times will be larger than that of SVM, and the time for training SAUC-SVM is comparable to that of AUC-SVM.

In conclusion, the proposed SAUC-SVM can achieve better AUC performance than the original AUC-SVM while still retaining comparable efficiency.

5. Conclusion

Since AUC is insensitive to imbalance class distribution and unequal misclassification cost [3,6,7,24,25], it is widely used as an alternative measure to accuracy [3,6,7]. Furthermore, algorithms specially designed to maximize accuracy may not lead to the best AUC performances [3], thus recently, AUC has been used as an optimization objective as well and lots of learning algorithms have respectively been developed to specially maximize AUC [1,2,8,14,18,26,27]. Among them, the large margin AUC-maximized classifier AUC-SVM [2,26] has attracted much attention. However, AUC-SVM usually emphasizes more on the local discriminative information just associated with these support vector *sample pairs* and hardly takes the overall view of sample pair set into account, thus may incur loss of global distribution information in the set of the sample pairs. Moreover, due to the high computation complexity of AUC-SVM incurred by the large number of training sample pairs quadratic in the number of samples, some sampling technique is usually adopted [2,8,26] but inevitably results in a further loss of the distribution information in the set of the sample pairs. In order to compensate the loss and simultaneously boost the performance of AUC-SVM, in this paper, we develop a novel structure-embedded AUC-SVM (SAUC-SVM) through embedding the structure information represented by the sample covariance matrix of the sample pair set (more specifically the difference vector set) into AUC-SVM, actually we can embed the equivalent sum of covariance matrices in individual classes for lower complexity. Through such an embedding, SAUC-SVM takes the distribution information in the whole data into

account and guarantees better generalization performance than AUC-SVM, as verified in both the toy and real experiments.

The high computation complexity for training AUC-SVM is still an open problem and the same problem also exists in our proposed SAUC-SVM, as a result, developing more efficient methods to reduce such complexity is still one of our future works. And we also plan to incorporate the structure information in data into the linear programming variants of AUC-SVM to boost their AUC performances.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and suggestions to significantly improve the presentation of this paper and National Science Foundations of China and Jiangsu Province under Grant Nos. 60773061, 6097XXXX and BK2008381 for support.

References

1. A. Herschtal and B. Raskutti, "Optimizing area under the ROC curve using gradient descent", in *Proceedings of the 21st International Conference on Machine Learning ICML'04*, Banff, Alberta, Canada (July 2004), pp. 49-56.
2. A. Rakotomamonjy, "Optimizing area under roc curve with SVMs", in *Proceedings of the 1st Workshop on ROC Analysis and Artificial Intelligence ROCAI'04*, Valencia, Spain (August 2004), pp. 71-80.
3. C. Cortes and M. Mohri, "AUC optimization vs. error rate minimization", *Advances in Neural Information Processing Systems NIPS'03,* Whistler, British Columbia, Canada (December 2003).
4. C. J. C. Burges, "A tutorial on support vector machines for pattern recognition", *Data Mining Knowledge Discovery*, **2** (1998), 121-167.
5. C. L. Blake and C. J. Merz, UCI repository of machine learning databases, 1998. Available at *http://archive.ics.uci.edu/ml/*

6. C. X. Ling, J. Huang and H. Zhang, "AUC: a better measure than accuracy in comparing learning algorithms", in *Proceedings of the 16th Canadian Conference on Artificial Intelligence AI'03*, Halifax**,** Nova Scotia**,** Canada (June 2003), pp. 329-341.
7. C. X. Ling, J. Huang and H. Zhang, "AUC: a statistically consistent and more discriminating measure than accuracy", in *Proceedings of International Joint Conference on Artificial Intelligence IJCAI'03*, Acapulco, Mexico (August 2003), pp. 519-526.
8. D. M. J. Tax and C. Veenman, "Tuning the hyperparameter of an AUC-optimized classifier", in *Proceedings of the Seventeenth Belgium-Netherlands conference on Artificial Intelligence BNAIC'05,* Brussels, Belgium (October 2005), pp. 224-231.
9. F. Y. Shih and K. Zhang, "Support vector machine networks for multi-class classification", *International Journal of Pattern Recognition and Artificial Intelligence*, **19** (2005), 775-786.
10. H. Xue, S. C. Chen and Q. Yang, "Structural Support Vector Machine", in *Proceedings of the 15th International Symposium on Neural Networks ISNN'08*, Beijing, China (September 2008), pp. 501-511.
11. H. Xue, S. C. Chen and Q. Yang, "Discriminatively regularized least-squares classification", *Pattern Recognition*, **41** (2009) 93-104.
12. H. Shin, "Neighborhood Property based Pattern Selection for Support Vector Machines", *Neural Computation*, **19** (2007) 816-855.
13. J. Shawe-Taylor and N. Cristianini, *Kernel methods for Pattern Analysis*. Cambridge University Press, 2004.
14. K. Ataman, W. N. Street and Y. Zhang, "Learning to rank by maximizing AUC with linear programming", in *IEEE International Joint Conference on Neural Networks IJCNN'06,* Vancouver, Canada (July 2006), pp. 123-129.
15. K. Huang, H. Yang, I. King and M. R. Lyu, "Learning large margin classifiers locally and globally", in *Proceedings of the 21st International Conference on Machine Learning ICML'04*, Banff, Alberta, Canada (July 2004), pp. 401-408.
16. K. Huang, H. Yang, I. King and M. R. Lyu, "Local learning vs. global learning: an introduction to maxi-min margin machine", in *Support Vector Machines: Theory and Applications*, eds. L. Wang, Springer, Berlin, 2005, pp. 113-131.

17. K. Huang, H. Yang, I. King and M. R. Lyu, "Maxi-Min Margin Machine: Learning Large Margin Classifiers Locally and Globally", *IEEE Transactions on Neural Networks*, **19** (2008) 260-272.
18. L. Yan, R. Dodier, M.C. Mozer and R. Wolniewicz, "Optimizing classifier performance via the Wilcoxon-Mann-Whitney statistic", in *Proceedings of the International Conference on Machine Learning ICML'03,* Washington D.C., USA (August 2003), pp. 848-855.
19. Mosek optimization software for solving the large-scale mathematical optimization problems, available at *http://www.mosek.com*.
20. M. A. Woodbury, "Inverting modified matrices", Technical Report 42, Statistical Research Group, Princeton University, Princeton, NJ, 1950
21. N. Crestianini and J.S. Taylor, *An introduction to Support vector Machines and other kernel-based learning methods,* Cambridge University Press, 2000.
22. N. Usunier, M. Amini and P. Gallinari, "A data-dependent generalization error bound for the AUC", in *Proceedings of the 22nd International Conference on Machine Learning Workshop on ROC Analysis in Machine Learning ROCML ICML'05,* Bonn, Germany (August 2005).
23. S. J. Kim and S. Boyd, "A minimax theorem with application to machine learning, signal processing, and finance", *SIAM Journal on Optimization*, **19** (2008) 1344-1367.
24. S. N. Srihari and H. Srinivasan, "Comparison of ROC and likelihood decision methods in automatic fingerprint verification", *International Journal of Pattern Recognition and Artificial Intelligence,* **22** (2008), 535-553.
25. T. Fawcett, "An introduction to ROC analysis", *Pattern Recognition Letters*, **27** (2006) 861-874.
26. U. Brefeld and T. Scheffer, "AUC maximizing Support Vector Machine learning", in *Proceedings of the 22nd International Conference on Machine Learning Workshop on ROC Analysis in Machine learning ROCML ICML'05,* Bonn, Germany (August 2005), pp. 377-384.

27. V.C. Raykar, R. Duraiswami and B. Krishnapuram, “A fast algorithm for learning a ranking function from large scale data sets”, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30** (2008) 1158-1170.