Published in final edited form as:
Pattern Recognit. 2015 January 1; 48(1): 276–287. doi:10.1016/j.patcog.2014.07.025.

Optimizing area under the ROC curve using semi-supervised learning

Shijun Wang[a], **Diana Li**[a], **Nicholas Petrick**[b], **Berkman Sahiner**[b], **Marius George Linguraru**[c,d], and **Ronald M. Summers**[a,*]

[a]Imaging Biomarkers and Computer-Aided Diagnosis Lab, Radiology and Imaging Sciences, National Institutes of Health Clinical Center, Bethesda, MD 20892-1182, United States

[b]Center for Devices and Radiological Health, U.S. Food and Drug Administration, Silver Spring, MD 20993, United States

[c]Sheikh Zayed Institute for Pediatric Surgical Innovation, Children's National Health System, Washington, DC 20010, United States

[d]School of Medicine and Health Sciences, George Washington University, Washington, DC 20037, United States

Abstract

Receiver operating characteristic (ROC) analysis is a standard methodology to evaluate the performance of a binary classification system. The area under the ROC curve (AUC) is a performance metric that summarizes how well a classifier separates two classes. Traditional AUC optimization techniques are supervised learning methods that utilize only labeled data (i.e., the true class is known for all data) to train the classifiers. In this work, inspired by semi-supervised and transductive learning, we propose two new AUC optimization algorithms hereby referred to as semi-supervised learning receiver operating characteristic (SSLROC) algorithms, which utilize unlabeled test samples in classifier training to maximize AUC. Unlabeled samples are incorporated into the AUC optimization process, and their ranking relationships to labeled positive and negative training samples are considered as optimization constraints. The introduced test samples will cause the learned decision boundary in a multidimensional feature space to adapt not only to the distribution of labeled training data, but also to the distribution of unlabeled test data. We formulate the semi-supervised AUC optimization problem as a semi-definite programming problem based on the margin maximization theory. The proposed methods SSLROC1 (1-norm) and SSLROC2 (2-norm) were evaluated using 34 (determined by power analysis) randomly selected datasets from the University of California, Irvine machine learning repository. Wilcoxon signed rank tests showed that the proposed methods achieved significant improvement compared

[1]Matlab code of the proposed methods will be released on http://clinicalcen ter.nih.gov/drd/summers.html once the paper is published.

[*]Correspondence to: Imaging Biomarkers and Computer-Aided Diagnosis Laboratory, Radiology and Imaging Sciences, National Institutes of Health Clinical Center, Building 10 Room 1C224D MSC 1182, Bethesda, MD 20892-1182, United States. Tel.: +1 301 402 5486; fax: +1 301 451 5721., rms@nih.gov (R.M. Summers)., URL: http://www.cc.nih.gov/about/SeniorStaff/ronald_summers.html (R.M. Summers).

Conflict of interest
Dr. Ronald Summers receives patent royalties and research support from iCAD.

with state-of-the-art methods. The proposed methods were also applied to a CT colonography dataset for colonic polyp classification and showed promising results.[1]

Keywords

Receiver operating characteristic; AUC; Semi-supervised learning; Transfer learning; Semidefinite programming; RankBoost; SVMROC; SSLROC

1. Introduction

Receiver operating characteristic (ROC) analysis is a standard methodology to evaluate the performance of a classification system [1–12]. It is applied extensively within clinical medicine [13–15]. The ROC curve is a two-dimensional plot which illustrates the relationship between the true positive rate (sensitivity) and the false positive rate (1 – specificity) of a binary classifier. In essence, a classifier seeks the optimal mapping of samples from a multi-dimensional feature space to a one-dimensional decision space during the training process. After the training process, the classifier can be applied to test samples whose labels are unknown and make a prediction for each test sample. The value of the prediction should be numerical (not binary categories) in order to make ROC analysis. Based on the predictions of the test set from a trained classifier, user of the classifier can select a specific diagnostic threshold to differentiate positive from negative samples for his or her specific application by finding the threshold along the ROC curve which maximizes sensitivity at the highest acceptable false positive rate (or cost).

The area under the ROC curve (AUC) is a univariate description of the ROC curve [1]. It ranges from 0.5 to 1, with larger values representing higher system performance. The AUC is equal to the probability that the decision value assigned to a randomly-drawn positive sample is greater than the value assigned to a randomly-drawn negative sample. Flach et al. proved that AUC is coherent and linearly related to expected loss [12]. The AUC statistic is commonly used to compare different classification systems. Previous studies have shown that AUC is statistically consistent and a more discriminative measure than classification accuracy [3,4].

Although some researchers have recommended the use of AUC for the evaluation of machine learning algorithms when a single performance metric needs to be used for the evaluation [1], others have pointed out some shortcomings of the use of the AUC. Lobo et al. cited a number of limitations of the use of AUC in evaluating the performance of species distribution (presence–absence) models [16] in ecology. Among the more general limitations are that the AUC summarizes performance over regions of the ROC space in which one would rarely operate, and that the goodness-of-fit of a model is ignored by the AUC. Hanczar et al. studied the problem of comparing estimates of AUC, true positive rate (TPR) and false positive rate (FPR) with true metrics when classifier training and performance estimation are performed on small-sample datasets [17]. They found that generally there is weak regression of the true metric on the estimated metric for all three figures of merit (AUC, TPR and FPR) studied. Clearly, AUC needs to be carefully considered as an endpoint in both classifier evaluation and classifier design. However, when a single figure of merit

needs to be used for classifier design, and the operating point of the classifier (a specific desired FPR or TPR) is not defined a priori, AUC remains a strong alternative to other figures of merit. AUC continues to be a very widely used endpoint in classifier evaluation and design, and many approaches to classifier design only indirectly maximize the AUC by optimizing some other cost functions, such as classification accuracy [18]. Our study does not try to define the scenarios for which AUC is an appropriate metric, but to instead discuss and compare approaches for optimizing AUC when it is deemed appropriate. Direct optimization of the AUC for a binary classifier is an interesting problem that may lead to improved performance for such applications.

In previous work, Rakotomamonjy first showed that support vector machines (SVMs) can maximize both AUC and accuracy [5]. He proposed a quadratic programming-based algorithm for AUC maximization by considering the margins between positive and negative training samples. Hereafter, we will refer this method as “SVMROC”. Subsequently, Brefeld and Scheffer presented a rigorous derivation of an AUC-maximizing SVM by imposing a convex bound and a margin item to the optimization problem [6]. They not only gave a strict analytical solution to the AUC-maximizing problem, but also showed an approximate solution based on clustering the constraints for large datasets.

Learning by an ensemble of classifiers is a very effective learning mechanism and a mainstream scheme used in machine learning [19,20]. Ensemble learning refers to a collection of methods that learn a target function by training a number of individual learners and combining their predictions together. Bagging [21] and boosting [22] are two of the best-known ensemble learning methods. Inspired by the “collaborative filtering” problem of ranking movies for a user based movie ratings from other users, Freund et al. proposed an efficient algorithm, termed RankBoost, for combining preferences based on the boosting approach [23]. RankBoost was originally designed for ranking problems. AUC optimization promotes ranks of positive training samples and decreases ranks of negative training samples, and is therefore essentially a ranking problem. RankBoost can thus be applied to AUC optimization, and has been widely used as a baseline method for this problem.

To maximize AUC for large scale and high dimensional data, Gao et al. proposed a one-pass AUC optimization technique called OPAUC [24]. The most prominent feature of this technique is that it only scans the data once as a single sequence and, therefore, does not require storage of the whole training set. OPAUC employs a square loss to measure the ranking error between two instances from different classes. A regression based algorithm was developed to calculate the first and second-order statistics of the training data and store them in memory. By this way, the storage requirement of OPAUC is only determined by the dimension of the data, not the number of instances of the data.

In recent years, semi-supervised learning (SSL) has emerged as an alternate approach to supervised learning in machine learning with advantages in many real life applications. Semi-supervised learning falls between supervised and unsupervised learning [25,26]. It utilizes both labeled data (usually a small amount), in which the true class is known, and unlabeled data (usually many), in which the data class is unknown, during the training process. Semi-supervised learning algorithms were developed primarily because the labeling

of data is typically expensive, and even impossible in some applications. It is especially useful for medical problems because the acquisition of labels is very expensive and time consuming for many clinical trials. Previous studies of semi-supervised learning focused on classification and clustering problems [25,26]. For classification problems, classification accuracy is a widely-used evaluation indicator to test semi-supervised learning methods.

Traditional AUC optimization techniques are supervised learning methods, which only utilize labeled data in classifier training. Previous studies on SSL have shown that by utilizing distribution or manifold information of test samples, SSL algorithms can achieve higher classification performance compared with supervised learning algorithms. Thus, one natural idea will be to apply the mechanism of SSL to the problem of AUC optimization. In addition, SSL also has a close connection to transductive learning. Traditional supervised learning algorithms attempt the difficult task of learning general rules from training data, but transductive learning reasons from observed training data to test cases directly [27,25]. This is quite different from traditional inductive learning, which only considers functions learned from a training set and ignores statistical connection between training and test sets. In transductive learning, an unlabeled test dataset is used during classifier training in order to predict class membership for the given test dataset based on the labels of training samples. Trans-ductive learning focuses on how to transfer the knowledge gained from the training samples to the unlabeled test samples in an efficient and accurate way. The motivation behind transductive learning is also applicable to the AUC optimization problem.

As an example of transductive learning, Sindhwani and Keerthi proposed semi-supervised linear support vector classifiers (named “SVMlin”) to handle partially-labeled large scale datasets with possibly very large and sparse features [28,29]. They applied modified finite Newton techniques to linear transductive SVM which is significantly more efficient and scalable than traditional dual optimization techniques for solving quadratic programming problems.

In the literature, there is little work on applying SSL or transductive learning to AUC optimization explicitly. Amini et al. proposed a boosting algorithm (“SSRankBoost”) for learning bipartite ranking functions with partially labeled data [30]. Bipartite ranking problem refers to a ranking problem which assigns higher scores to relevant examples than to irrelevant ones for a given dataset which has wide applications in document analysis area. Along the same line, Ralaivola proposed a semi-supervised bipartite ranking algorithm with the normalized Rayleigh coefficient [31]. Later, Usunier et al. proposed a multiview semi-supervised learning algorithm for ranking multilingual documents [32]. Since AUC optimization has close relationship with ranking problem, works on learning bipartite ranking functions can also be applied to AUC optimization problems.

In this work, inspired by semi-supervised and transductive learning, we propose two new AUC optimization algorithms hereby referred to as semi-supervised learning receiver operating characteristic algorithms (SSLROC1 and SSLROC2), which utilize unlabeled test samples for classifier training. Unlabeled test samples are incorporated into the AUC optimization process, and their ranking relationships to positive and negative training samples are considered as optimization constraints. The introduced test samples make the

learned decision boundary in a multi-dimensional feature space to adapt not only to the distribution of labeled training data, but also to the distribution of unlabeled test data. We formulate the semi-supervised AUC optimization problem as a semi-definite programming (SDP) problem [33] based on the margin maximization theory.

The paper is organized as follows: we first introduce the AUC optimization problem in Section 2. The AUC optimization problem is then formulated as a semi-supervised learning problem based on the margin maximization theory and solved using semi-definite programming in Section 3. In Section 4, we list 34 datasets (from University of California, Irvine machine learning repository) which are used to evaluate the proposed method, and show comparisons with state-of-the-art classification or AUC optimization methods. We also show results from the proposed method for a colonic polyp classification problem based on a biomedical imaging dataset. In Section 5 we conclude our findings and discuss computational complexity issues and future research directions.

2. Maximizing AUC with large margin learning

For a two-class classification problem, given training samples $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, $y_i \in \{-1, +1\}$, the optimization problem for maximizing the area under the ROC curve is defined as follows:

Optimization problem 1

$$\max AUC = \max_{\boldsymbol{w}} \frac{\sum_{i=1}^{n^+}\sum_{j=1}^{n^-} I(\xi_{ij}>0)}{n^+ \times n^-} \quad (1)$$

with $\xi_{ij} = \langle \boldsymbol{w}, \phi(x_i^+)\rangle - \langle \boldsymbol{w}, \phi(x_j^-)\rangle$, $i = 1, 2, \ldots, n^+$, $j = 1, 2, \ldots, n^-$, where n^+ and n^- are the numbers of positive (+1) and negative (−1) training samples, respectively; φ: $X \rightarrow F$ denotes a mapping function which maps the input space X into a new feature space F; $\boldsymbol{w}$ is the weight of the linear classifier; I is the indicator function (1: when condition holds; 0: otherwise). The key idea of above formulation of AUC maximization is to assign higher prediction values for positive training samples compared with negative training samples and make the learned classifier work for as many positive–negative sample pairs as possible.

Since optimization problem 1 is not differentiable, Rakotomamonjy proposed the following approximately equivalent problem (1-norm or 2-norm) based on a large margin learning theory [5]:

Optimization problem 2

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n^+}\sum_{j=1}^{n^-}\xi_{ij}^r \quad (2)$$

with

$$\begin{gathered}\langle \boldsymbol{w}, \phi(x_i^+)\rangle - \langle \boldsymbol{w}, \phi(x_j^-)\rangle \geq 1 - \xi_{ij}, \\ \xi_{ij} \geq 0, \\ i = 1, 2, \ldots, n^+, j = 1, 2, \ldots, n^-, \\ r \in \{1, 2\}.\end{gathered}$$

The above constrained quadratic programming optimization problem 2 can be solved using the Lagrange multiplier optimization method [5]. Optimization problem 2 attempts to identify a linear classifier in the reproducing kernel Hilbert space, which makes correct predictions for every positive–negative pair in the training set with certain relaxation $\xi_{ij} \geq 0$, $i = 1, 2, \ldots, n^+$, $j = 1, 2, \ldots, n^-$.

3. A semi-supervised learning method for AUC optimization

In optimization problem 2 we consider only training samples during the AUC optimization process. Therefore, this is a supervised learning algorithm in essence. It has been shown in the semi-supervised learning literature that adding information from unlabeled test samples can be helpful in identifying a more accurate decision boundary in classification problems [25,26]. One natural question is how to best utilize the information contained in the unlabeled test set to help maximize the AUC during optimization in large margin leaning classifiers (e.g., SVMs).

To extend large margin learning to the semi-supervised learning domain, Bennett and Demiriz proposed a semi-supervised support vector machine (S^3VM) [34]. S^3VM minimizes both the classification accuracy and the function capacity based on available data in both training and test sets. The key idea in the formulation of S^3VM is the incorporation of unlabeled test sample constraints within the large margin learning framework. Because the labels for the test samples are unknown, two constraints are imposed in the optimization problem for each test sample. This corresponds to the situation in which the unknown test sample is first assumed to be a positive sample, and then a negative sample. Later, Sindhwani and Keerthi proposed semi-supervised linear SVMs to handle large scale data [28,29].

Inspired by the above-mentioned work on semi-supervised SVMs, in this paper we proposed two new semi-supervised algorithms to solve the AUC optimization problem 2. The basic idea is to incorporate unlabeled data in the AUC optimization framework shown in problem 2 and guess labels of unlabeled data during the optimization process. For each test sample, we first assume it is positive and compare it with all negative training samples; then we assume it is negative and compare it with all positive training samples. By this way, we hope we can rank potential positive samples higher compared with potential negative samples in the test set with the guidance of labeled training samples. In another words, here we propose to utilize unlabeled test data which is the essence of semi-supervised learning and try to rank as many positive test samples higher (compared with negative samples) as possible which is the essence of AUC optimization. More specifically, for a two-class classification problem, given positive training samples $\{(x_1^+, y_1^+), \ldots, (x_p^+, y_p^+)\}$, $y_i^+ \in \{+1\}$ $i = 1, 2, \ldots, p$, negative training samples $\{(x_1^-, y_1^-), \ldots, (x_q^-, y_q^-)\}$, $y_j^- \in \{-1\}$, $j = 1, 2, \ldots, q$, and test

samples $\{(x_1), \ldots, (x_r)\}$ without labels, the optimization problem for maximizing the AUC under the semi-supervised learning settings is defined as

Optimization problem 3 (1-norm)

$$\begin{aligned}\min_{\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{u},\boldsymbol{d}} \tfrac{1}{2}\|\boldsymbol{w}\|^2+\tfrac{C_1}{2}\sum_{i=1}^{p}\sum_{j=1}^{q}\xi_{ij}+\tfrac{C_2}{2}\sum_{m=1}^{r}\left(\sum_{j=1}^{q}\eta_{mj}+\sum_{i=1}^{p}\mu_{mi}\right)\\ \text{s.t.}\quad (\langle \boldsymbol{w},\phi(x_i^+)\rangle-\langle \boldsymbol{w},\phi(x_j^-)\rangle)\geq 1-\xi_{ij},\\ \xi_{ij}\geq 0,\quad i=1,2,\ldots,p,\ j=1,2,\ldots,q,\\ (\langle \boldsymbol{w},\phi(x_m)\rangle-\langle \boldsymbol{w},\phi(x_j^-)\rangle)+M(1-d_m)\geq 1-\eta_{mj},\\ \eta_{mj}\geq 0,\quad m=1,2,\ldots,r,\ j=1,2,\ldots,q,\\ -1\times(\langle \boldsymbol{w},\phi(x_m)\rangle-\langle \boldsymbol{w},\phi(x_i^+)\rangle)+Md_m\geq 1-\mu_{mi},\\ \mu_{mi}\geq 0,\quad m=1,2,\ldots,r,\ i=1,2,\ldots,p,\end{aligned} \tag{3}$$

where $\boldsymbol{w}$ is the linear classifier to be identified; margin size parameter M is a sufficiently large constant introduced to handle the margins between test samples and positive/negative training samples; C_1 and C_2 are trade-off parameters to balance classifier complexity, training error of training samples, and impact from unlabeled test samples; $\xi_{ij} \geq 0$, $i = 1, 2, \ldots, p$, $j = 1, 2, \ldots, q$, are margins introduced to accommodate non-linear separable positive–negative pairs in the training set; $\eta_{mj} \geq 0$, $m = 1, 2, \ldots, r$, $j = 1, 2, \ldots, q$, are margins introduced for test-negative sample pairs; $\mu_{mi} \geq 0$, $m = 1, 2, \ldots, r$, $i = 1, 2, \ldots, p$, are margins introduced for test-positive sample pairs. $d_m \in \{0, 1\}$, $m = 1, 2, \ldots, r$, are estimated labels of the unlabeled test samples (0 means negative sample). The objective function shown in Eq. (3) contains three parts: the first part is a penalty item on the complexity of the classifier; the second part weighted by C_1 contains training errors on positive–negative pairs; the last part weighted by C_2 deals with the empirical errors from unlabeled test data. In the constraints shown above, there are also three parts: the first part shows the pair-wise empirical errors from the training set; the second/third part shows empirical errors when compare test samples with negative/positive training samples (labels of test samples are estimated during the optimization process). Based on d_m and M, for each test sample, although there are two constraints in Eq. (3), actually only one constraint will take effect. In our experiments, we kept C_1 and C_2 equal to make the algorithm simple. A key advance in this approach is the inclusion of manifold information of test samples as part of the AUC maximizing (or ranking) constraints regarding positive/negative training samples.

Theorem 1—The optimal solution for quadratic optimization problem 3 can be found by solving the following semidefinite programming (SDP) problem:

$$\begin{aligned}\min_{\boldsymbol{d_3},\boldsymbol{\psi},\boldsymbol{\zeta}}\ & t\\ \text{s.t.}\quad & \begin{bmatrix}\boldsymbol{K} & \frac{(\boldsymbol{d_3}-\boldsymbol{\psi}+\boldsymbol{\zeta})}{\sqrt{2}}\\ \frac{(\boldsymbol{d_3}-\boldsymbol{\psi}+\boldsymbol{\zeta})^T}{\sqrt{2}} & t-\boldsymbol{\psi}^T\boldsymbol{e_3}\end{bmatrix}\geq 0,\\ & \boldsymbol{\psi}\geq 0,\quad \boldsymbol{\zeta}\geq 0,\end{aligned} \tag{4}$$

where

$$\boldsymbol{d_3}=\begin{pmatrix} \boldsymbol{e} \\ -\boldsymbol{d}^N \\ -\boldsymbol{d}^P \end{pmatrix},$$
$$d^N_{(m-1)*q+j}=(M(1-d_m)-1),\quad m=1,2,\ldots,r,\quad j=1,2,\ldots,q,$$
$$d^P_{(m-1)*p+i}=(Md_m-1),m=1,2,\ldots,r,i=1,2,\ldots,p,$$
$$K^{PNPN}_{i_1j_1,i_2j_2}=k_{i_1i_2}-k_{i_1j_2}-k_{j_1i_2}+k_{j_1j_2},$$
$$K^{PNUN}_{ij_1,mj_2}=k_{im}-k_{ij_2}-k_{j_1m}+k_{j_1j_2},$$
$$K^{PNUP}_{i_1j,mi_2}=k_{i_1m}-k_{i_1i_2}-k_{jm}+k_{ji_2},$$
$$K^{UNUN}_{m_1j_1,m_2j_2}=k_{m_1m_2}-k_{m_1j_2}-k_{j_1m_2}+k_{j_1j_2},$$
$$K^{UNUP}_{m_1j,m_2,i}=k_{m_1m_2}-k_{m_1i}-k_{jm_2}+k_{ji},$$
$$K^{UPUP}_{m_1i_1,m_2i_2}=k_{m_1m_2}-k_{m_1i_2}-k_{i_1m_2}+k_{i_1i_2},$$

$k_{ij} = \langle \varphi(x_i), \varphi(x_j) \rangle$, $\varphi(\mathrm{x})$ is a chosen kernel function, x_i and x_j are samples from the set denoted by the corresponding superscripts, $\boldsymbol{K}^{PNUP} = \boldsymbol{K}^{UPPN^T}$, $\boldsymbol{K}^{PNUN} = \boldsymbol{K}^{UNPN^T}$, $\boldsymbol{K}^{UNUP} = \boldsymbol{K}^{UPUN^T}$ and

$$K=\begin{bmatrix} +\boldsymbol{K}^{PNPN}+\boldsymbol{K}^{PNUN}-\boldsymbol{K}^{PNUP} \\ +\boldsymbol{K}^{UNPN}+\boldsymbol{K}^{UNUN}-\boldsymbol{K}^{UNUP} \\ -\boldsymbol{K}^{UPPN}-\boldsymbol{K}^{UPUN}+\boldsymbol{K}^{UPUP} \end{bmatrix}.$$

The proof of this theorem is shown in Appendix A. In the above definitions, *P* means positive training samples; *N* means negative training samples; *U* means unknown test samples. Each block in the block matrix **K** contains kernel function values from four datasets denoted by its superscript.

The AUC optimization problem using semi-supervised learning can also be formulated using 2-norm soft margin:

Optimization problem 4 (2-norm)

$$\begin{gathered} \min_{\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{d}} \frac{1}{2}\|\boldsymbol{w}\|^2+\frac{C_1}{2}\sum_{i=1}^{p}\sum_{j=1}^{q}\xi_{ij}^2+\frac{C_2}{2}\sum_{m=1}^{r}\left(\sum_{j=1}^{q}\eta_{mj}^2+\sum_{i=1}^{p}\mu_{mi}^2\right) \\ \text{s.t.}\quad (\langle \boldsymbol{w},\phi(x_i^+)\rangle-\langle \boldsymbol{w},\phi(x_j^-)\rangle)\geq 1-\xi_{ij} \\ \xi_{ij}\geq 0,\quad i=1,2,\ldots,p,\ j=1,2,\ldots,q \\ (\langle \boldsymbol{w},\phi(x_m)\rangle-\langle \boldsymbol{w},\phi(x_j^-)\rangle)+M(1-d_m)\geq 1-\eta_{mj} \\ \eta_{mj}\geq 0,\quad m=1,2,\ldots,r,\ j=1,2,\ldots,q \\ -1\times(\langle \boldsymbol{w},\phi(x_m)\rangle-\langle \boldsymbol{w},\phi(x_i^+)\rangle)+Md_m\geq 1-\mu_{mi} \\ \mu_{mi}\geq 0,\quad m=1,2,\ldots,r,\ i=1,2,\ldots,p \end{gathered} \tag{5}$$

Theorem 2—The optimal solution for quadratic optimization problem 4 can be found by solving the following SDP problem:

$$\min_{\boldsymbol{d_3},\boldsymbol{\zeta}} \quad \text{s.t.} \quad \begin{bmatrix} \boldsymbol{K} & \frac{(\boldsymbol{d_3}+\boldsymbol{\zeta})}{\sqrt{2}} \\ \frac{(\boldsymbol{d_3}+\boldsymbol{\zeta})^T}{\sqrt{2}} & t \end{bmatrix} \geq 0, \quad \boldsymbol{\zeta} \geq 0,$$

where

$$\boldsymbol{d_3} = \begin{pmatrix} \boldsymbol{e} \\ -\boldsymbol{d}^N \\ -\boldsymbol{d}^P \end{pmatrix},$$

and

$$K = \begin{bmatrix} +\boldsymbol{K}^{PNPN} + \boldsymbol{K}^{PNUN} - \boldsymbol{K}^{PNUP} \\ +\boldsymbol{K}^{UNPN} + \boldsymbol{K}^{UNUN} - \boldsymbol{K}^{UNUP} \\ -\boldsymbol{K}^{UPPN} - \boldsymbol{K}^{UPUN} + \boldsymbol{K}^{UPUP} \end{bmatrix}.$$

The proof of Theorem 2 is similar to Theorem 1.

4. Experimental validation

4.1. Experimental settings

To evaluate the proposed SSLROC1 (1-norm) and SSLROC2 (2-norm) AUC optimization methods, we compared them with SVMs [35] and three state-of-the-art supervised AUC optimization methods: SVMROC [5], RankBoost [23] and OPAUC [24]. We also compare the proposed methods with two semi-supervised classifiers SSRank-Boost [30] and SVMlin [28,29] to show the advantages unlabeled data bring to the AUC optimization problem. For each tested method and dataset we used 5 × 2-fold cross validation, which contains 5 repetitions of 2-fold cross validation (CV). The validation method was inspired by Dietterich's 5 × 2 CV paired *t*-test study [36], which has a low probability of incorrectly detecting a difference when no difference exists (type-I error), and a reasonable probability of detecting a difference when it exists (power). We calculated the AUC for each test fold of the 5 × 2-fold CV using prediction values from each method, and used the AUC average of the ten test folds to evaluate the performance of each method on each dataset. To determine whether two compared methods have a significant difference across multiple datasets, we used a Wilcoxon paired signed rank test.

For all datasets, we used *z*-score to normalize all features available such that each feature is centered to have mean of zero and scaled to have standard deviation of one. For SVMs, SVMROC, SSLROC1 and SSLROC2 (the four kernel based learning methods), we used a Gaussian radial basis function (RBF) as a kernel function for the similarity calculation, and the width factor σ was set as the 90th percentile of pairwise distances (in ascending order) between all instances or samples for each dataset. For SVMs and SVMROC, the classifier

complexity and training error trade off parameter C was varied from 1×10^{-4} to 1×10^{2}, increasing linearly in log 10 scale. For RankBoost, we tuned the number of weak learners from 30 to 90 to identify the optimal parameter. We explored the same parameter space for OPAUC as the authors proposed in ref. [24] to identify the optimal parameter combinations: learning rate η changes from 2^{-12} to 2^{10} and regularization parameter λ changes from 2^{-10} to 2^{2} (change linearly in log 2 scale). For SVMlin [28,29], we tested the following parameter combinations: regularization parameter λ [10^{-4}, 10^{4}] and λ_u [10^{-2}, 10^{2}] (linearly changing in the log 10 scale). The parameters used for SSRankBoost [30] are discount factor [0:0.2:1] and the number of unlabeled examples K [1:10]. For the proposed methods SSLROC1 and SSLROC2, trade off parameter C was set from 10^{-3} to 10^{1} (change linearly in log 10 scale) and margin size parameter M was set as one of the three values: 0.1, 1 and 10.

Matlab was used as the programming environment in this study. We employed the public open source Matlab toolboxes SDPT3 [37], Sedumi [38] and YALMIP [39] as the SDP solver. For SVMROC and RankBoost we employed the SVM-KM kernel learning toolbox [40] (http://asi.insa-rouen.fr/~arakotom/toolbox/index). SVMlin was downloaded from http://vikas.sindhwani.org/svmlin.html. OPAUC was from Prof. Zhihua Zhou's lab (http://lamda.nju.edu.cn). SSRankBoost was downloaded from http://ama.liglab.fr/~amini/SSRankBoost/.

4.2. Experimental results on UCI datasets

4.2.1. UCI datasets—To test the proposed ROC optimization algorithm and compare it with SVMs and traditional ROC optimization methods, we employed the University of California, Irvine (UCI) machine learning repository [41]. The UCI machine learning repository contains more than 200 datasets contributed from various application domains and is widely used in the machine learning community to evaluate various algorithms such as clustering, feature extraction, classification, and regression.

To determine the number of datasets needed for the experiments, we performed power analysis [42] using a Wilcoxon paired signed rank test. Power analysis showed that 34 datasets were needed in order to secure a 10% probability of getting a type I error and a 20% probability of getting a type II error (alpha=0.1, power=0.8) for the comparison between the proposed method and SVMs. Thus, we randomly selected 34 classification datasets from the UCI Repository. Note, we did not account for multiple hypotheses in our sample size calculation. All datasets had an attribute that could be used as a class label. Some were multi-class classification problems converted to binary classification problems based on previous published work using these datasets. In Table 1 we list all datasets used in this study along with the number of instances, attributes, and class label for each dataset. Due to computational considerations, we randomly selected 100 instances or samples from each dataset if it contained more than 100 instances.

4.2.2. Results—In Table 2 we show the average AUC of the eight compared methods tested on each of the 34 UCI datasets using the 5 × 2-fold CV. For each dataset and each method tested, the AUC shown was from the optimal parameters which achieved the highest AUC performance. We also show the corresponding standard deviation for each method on

each dataset. Standard deviation was calculated based on the ten AUC values from 5 × 2-fold CV. In Table 3 we list the numbers of win-tie-loss between the eight methods (pairwise) on the 34 UCI datasets. We observed that compared with state-of-the-art classification methods, SSLROC1 and SSLROC2 showed superior performance on more datasets. In Table 4 we show p values of the Wilcoxon signed rank tests between the eight methods (pairwise) on the 34 UCI datasets. Since the highest p-value is less than α=0.05, Hochberg's method for multiple tests of statistical significance [43] indicates that SSLROC1 and SSLROC2 have significantly improved performance compared with other methods. Also from the table we find that the difference between the proposed methods SSLROC1 (1-norm) and SSLROC2 (2-norm) does not reach statistical significance.

For the proposed SSLROC1 and SSLROC2 methods, there are two critical parameters which control their generalization ability. They are training error trade-off parameter C and margin size parameter M. To identify the influence of C and M on the performance of the proposed methods, in Fig. 1 we show the average AUC of SSLROC1 and SSLROC2 on three example UCI datasets when different C and M were used in the experiment. From these four example cases, we can find a trend in the parameter combinations which leads to better performance. To reduce computation load, we only explored a small parameter space spanned by M and C. There were 15 combinations of them in total which are few. For example, in the work of OPAUC, the authors tested a parameter space spanned by the learning rate eta (2^{-12}–2^{10}) and regularization parameter lambda (2^{-10}–2^{2}), 299 parameter combinations in total. From the trends shown on the four UCI datasets, we see that there is a high probability that exploring a larger parameter space will lead to better AUC performance.

4.3. Experimental results on CTC dataset

Colorectal cancer is the second-leading cause of cancer death in Americans [44]. Computed tomographic colonography (CTC), also known as virtual colonoscopy, provides a less invasive alternative to optical colonoscopy in screening patients for colonic polyps [45]. In Fig. 2, we show 3D volume rendering of a segmented colon and a typical colonic polyp on the fold. Previous studies showed that computer-aided detection systems can assist radiologists in CTC reading and improve their detection performance [46–49]. To show the effectiveness of our proposed methods and their potential applications in the CTC computer-aided detection system (CAD), we tested all four methods on a CTC dataset and analyzed the results using ROC analysis.

4.3.1. CTC datasets—Our dataset consisted of CTC examinations of 50 patients collected from three medical centers. Each patient had one or more polyps ≥6 mm confirmed by histopathological evaluation following optical colonoscopy (OC). Each patient was scanned in the supine and prone positions, and each scan was performed during a single breath hold using a 4- or 8-channel CT scanner. CT scanning parameters included 1.25- to 2.5-mm section collimation, 15 mm/s table speed, 1-mm reconstruction interval, 100 mAs, and 120 kVp. For each CT scan in the dataset, we segmented the colon first from the original 3D image [50]. Then we searched the inner surface of the colon to identify initial colonic polyp candidates. Our initial detection scheme based on surface curvature analysis reported 60

colonic polyps 5–30 mm in size and 5234 false positives. The labels of initial detections were determined by OC examination which is a golden standard in CTC. Each initial detection defined as a CAD detection represents a candidate polyp. After initial detection we extracted 157 3D geometric features from each colonic polyp candidate [47]. The polyps were confirmed by traditional optical colonoscopy. To make the problem computationally feasible we filtered the initial dataset to 100 CAD detections, which included 49 true detections and 51 false positives by removing true and false positives with low SVM vote values predicted by a SVM committee classifier [51]. 5 × 2-fold CV was performed on the filtered dataset and test set in CV was treated as unlabeled samples under our SSL learning framework.

4.3.2. Results—In Fig. 3, we show AUCs of eight methods on the CTC dataset. RankBoost showed the highest performance with an AUC of 0.914. The proposed SSLROC2 method was ranked as the second highest performance with AUC of 0.909. Please note that both SSLROC1 and SSLROC2 outperformed all other semi-supervised learning methods for AUC maximization. In Fig. 4, we show comparisons of SSLROC1 and SSLROC2 with different parameters C and M. They both achieved highest performance when $\log 10(C) = -1$ and $\log 10(M) = 0$.

5. Discussion and conclusion

We proposed two new AUC optimization methods called SSLROC1 and SSLROC2, which introduce test samples in the optimization of margins in a binary classification problem for the purpose of AUC maximization. We tested the proposed methods on 34 randomly selected UCI machine learning datasets. The SSLROC algorithms were found to have superior AUCs in a significantly larger fraction of UCI datasets compared with SVMs, SVMROC, RankBoost, OPAUC, SVMlin, and SSRankBoost which are state-of-the-art classification and AUC optimization methods. The proposed methods also showed advantages in a colonic polyp classification problem for a dataset of CT colonography cases compared with other methods except RankBoost.

SVMs have a complexity of $O(kn^2)$ for RBF kernels and $O(kn)$ for linear kernels, where n and k are the number of training samples and features, respectively. For our proposed method the computational complexity will increase to $O(25kn^4/16)$ and $O(25kn^2/16)$ for RBF and linear kernels, respectively, due to the introduction of test samples during AUC optimization. Here we assume that the training and testing sets have the same number of instances, which is the case for 5 × 2-fold cross validation; we also assume that the number of positive and negative samples is equal. For SVMROC, the computational complexity are $O(kn^4/4)$ and $O(kn^2/4)$ for RBF and linear kernels, respectively, under the assumption that the number of positive and negative samples is equal. The computational complexity analysis shows that the computational complexity is two orders of magnitude higher for both SVMROC and SSLROC over SVMs. For this reason the proposed method was applied to only small datasets in our study. However, the increased complexity of our method is balanced by its significantly higher performance over the other techniques. In future work we will investigate how to develop a more computationally efficient algorithm, likely using

more efficient algorithms to approximate the solution of the AUC maximization problem in large datasets [8].

Another potential disadvantage of the SSLROC method (and all transductive learning algorithms) is that when a new test dataset is acquired, the algorithm needs to be re-trained using the new test set as unlabeled data. This is in contrast to inductive learning algorithms (including all supervised algorithms), where the trained classifier can be directly applied to a new test dataset. In the field of computer-aided detection and diagnosis for radiological images, it is preferred to have a well trained CAD system and deploy it to hospitals or clinics without further training. Thus, a future research topic of interest will be to combine online and transductive learning to address the retraining issue in transductive AUC learning.

As we showed in the previous section, SSLROC1 and SSLROC2 did not reach statistical significance on the 34 UCI datasets. In the literature, Ng showed that sample complexity which is the minimum number of training examples required to train a good classifier grows only logarithmically as the number of irrelevant features increases in the dataset when L1 regularization is employed [52]; L2 regularization has a worst sample complexity that grows at least linearly. In the work of Zhu et al. on 1-norm SVMs, they also argue that 1-norm SVM has some advantages over 2-norm SVM when data contains redundant noise features [53]. For the proposed methods, the major difference is that we use different norms for the regularization. So based on studies shown above, SSLROC1 should beat SSLROC2 when data contain irrelevant noisy features. However, from experimental results shown in Table 2, we did not observe such kind of trend when we compare average AUCs of SSLROC1 and SSLROC2. We suspect that it might be related with small size data employed in this study. In the future, it will be interesting to investigate how the data size affects the generalization performance of the two proposed methods.

In conclusion, we developed new methods of AUC optimization based on semi-supervised learning and transductive learning that yield improved classifier performance on multiple public datasets. The proposed methods may lead to improved classification performance in diverse realms of data analysis including medical imaging and computer vision.

Acknowledgments

This work was supported by the Intramural Research Programs of the NIH Clinical Center and the Food and Drug Administration. We thank the NIH Biowulf computer cluster and Ms. Sylvia Wilkerson for their support on parallel computations. No official endorsement by the National Institutes of Health or the Food and Drug Administration of any equipment or product of any company mentioned in the publication should be inferred.

References

1. Bradley AP. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognit. 1997; 30:1145–1159.
2. Hand DJ, Till RJ. A simple generalisation of the area under the ROC curve for multiple class classification problems. Mach Learn. 2001; 45:171–186.
3. Ling, CX.; Huang, J.; Zhang, H. AUC: a statistically consistent and more discriminating measure than accuracy. 18th International Joint Conferences on Artificial Intelligence (IJCAI ‘03); 2003.

4. Cortes C, Mohri M. AUC optimization vs. error rate minimization. Advances in Neural Information Processing. 2004
5. Rakotomamonjy, A. Optimizing area under ROC curves with SVMs. ECAI 04 ROC and Artificial Intelligence Workshop; 2004.
6. Brefeld, U.; Scheffer, T. AUC maximizing support vector learning. Workshop on ROC Analysis in Machine Learning; 2005.
7. Huang J, Ling CX. Using AUC and accuracy in evaluating learning algorithms. IEEE Trans Knowl Data Eng. 2005; 17:299–310.
8. Calders T, Jaroszewicz S. Efficient AUC optimization for classification. Knowledge Discovery in Databases: PKDD 2007. 2007; 4702
9. Lee WH, Gader PD, Wilson JN. Optimizing the area under a receiver operating characteristic curve with application to land-mine detection. IEEE Trans Geosci Remote Sens. 2007; 45:389–397.
10. Vanderlooy S, Hullermeier E. A critical analysis of variants of the AUC. Mach Learn. 2008; 72:247–262.
11. Toh KA, Kim J, Lee S. Maximizing area under ROC curve for biometric scores fusion. Pattern Recognit. 2008; 41:3373–3392.
12. Flach, P.; Hernández-Orallo, J.; Ferri, C. A coherent interpretation of AUC as a measure of aggregated classification performance. International Conference on Machine Learning; 2011.
13. Zweig MH, Campbell G. Receiver-operating characteristic (ROC) plots—a fundamental evaluation tool in clinical medicine. Clin Chem. 1993; 39(4):561–577. [PubMed: 8472349]
14. Wang, S.; McKenna, M.; Petrick, N.; Sahiner, B.; Linguraru, MG.; Wei, Z.; Yao, J.; Summers, RM. ROC-like optimization by sample ranking: application to CT colonography. 2012 9th IEEE International Symposium on Biomedical Imaging (ISBI); 2012. p. 478-481.
15. Berrar D, Flach P. Caveats and pitfalls of ROC analysis in clinical microarray research (and how to avoid them). Brief Bioinform. 2012; 13(1):83–97. [PubMed: 21422066]
16. Lobo JM, Jiménez-Valverde A, Real R. AUC: a misleading measure of the performance of predictive distribution models. Glob Ecol Biogeogr. 2007; 17(2):145–151.
17. Hanczar B, Hua J, Sima C, Weinstein J, Bittner M, Dougherty ER. Small-sample precision of ROC-related estimates. Bioinformatics. 2010; 26(6):822–830. [PubMed: 20130029]
18. Marrocco C, Duin RPW, Tortorella F. Maximizing the area under the ROC curve by pairwise feature combination. Pattern Recognit. 2008; 41(6):1961–1974.
19. Liu Y, Yao X. Ensemble learning via negative correlation. Neural Netw. 1999; 12(10):1399–1404. [PubMed: 12662623]
20. Dietterich TG. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, and randomization. Mach Learn. 2000; 40(2):139–157.
21. Breiman L. Bagging predictors. Mach Learn. 1996; 24(2):123–140.
22. Freund, Y.; Schapire, RE. Computational Learning Theory. Vol. 904. Springer; Berlin/Heidelberg: 1995. A decision-theoretic generalization of on-line learning and an application to boosting; p. 23-37.
23. Freund Y, Iyer R, Schapire R, Singer Y. An efficient boosting algorithm for combining preferences. J Mach Learn Res. 2003; 4:933–969.
24. Gao, W.; Jin, R.; Zhu, S.; Zhou, Z-H. One-pass AUC optimization. 30th International Conference on Machine Learning; 2013.
25. Chapelle, O.; Scholkopf, B.; Zien, A. Semi-supervised Learning. MIT Press; Cambridge, MA, USA: 2006.
26. Zhu, X. Technical Report. University of Wisconsin; Madison: 2007. Semi-supervised learning literature survey.
27. Gammerman, A.; Vovk, V.; Vapnik, V. Learning by transduction. Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI); 1998. p. 148-155.
28. Sindhwani, V.; Keerthi, SS. Large scale semi-supervised linear SVMs. The 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 2006. p. 477-484.

29. Sindhwani, V.; Keerthi, SS. Large Scale Kernel Machines. MIT Press; Cambridge MA, US: 2007. Newton methods for fast solution of semi-supervised linear SVMs.
30. Amini M-R, Truong T-V, Goutte C. A boosting algorithm for learning bipartite ranking functions with partially labeled data. ACM Special Interest Group on Information Retrieval (ACM SIGIR). 2008:99–106.
31. Ralaivola, L. Semi-supervised bipartite ranking with the normalized Rayleigh coefficient. European Symposium on Artificial Neural Networks—Advances in Computational Intelligence and Learning; 2009. p. 47-52.
32. Usunier, N.; Amini, M-R.; Goutte, C. Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science. Vol. 6913. Springer; New York City, US: 2011. Multiview semi-supervised learning for ranking multilingual documents; p. 443-458.
33. Vandenberghe L, Boyd S. Semidefinite programming. SIAM Rev. 1996; 38(1):49–95.
34. Bennett, K.; Demiriz, A. Semi-supervised support vector machines. In: Michael, DAC.; Kearns, S.; Solla, Sara A., editors. Advances in Neural Information Processing Systems. Vol. 11. 1999. p. 368-374.
35. Chang CC, Lin CJ. LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol. 2011; 2(3):27:1–27:27.
36. Dietterich TG. Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput. 1998; 10(7):1895–1923. [PubMed: 9744903]
37. Tutuncu RH, Toh KC, Todd MJ. Solving semidefinite-quadratic-linear programs using SDPT3. Math Progr. 2003; 95(2):189–217.
38. Labit, Y.; Peaucelle, D.; Henrion, D. SEDUMI INTERFACE 1.02: a tool for solving LMI problems with SEDUMI. Proceedings of IEEE International Symposium on Computer Aided Control System Design; 2002. p. 272-277.
39. Lofberg, J. YALMIP: a toolbox for modeling and optimization in MATLAB. 2004 IEEE International Symposium on Computer Aided Control Systems Design; 2004.
40. Canu, S.; Grandvalet, Y.; Guigue, V.; Rakotomamonjy, A. SVM and Kernel Methods Matlab Toolbox. Perception Systemes et Information, INSA de Rouen; Rouen, France. 2005.
41. Frank, A.; Asuncion, A. UCI machine learning repository. University of California, School of Information and Computer Science; Irvine, CA: <http://archive.ics.uci.edu/ml>
42. Lachin JM. Introduction to sample-size determination and power analysis for clinical-trials. Control Clin Trials. 1981; 2(2):93–113. [PubMed: 7273794]
43. Hochberg Y. A sharper Bonferroni procedure for multiple tests of significance. Biometrika. 1988; 75(4):800–802.
44. Siegel R, Naishadham D, Jemal A. Cancer statistics 2012. Ca-a Cancer J Clin. 2012; 62(1):10–29.
45. Pickhardt PJ, Choi JR, Hwang I, Butler JA, Puckett ML, Hildebrandt HA, Wong RK, Nugent PA, Mysliwiec PA, Schindler WR. Computed tomographic virtual colonoscopy to screen for colorectal neoplasia in asymptomatic adults. N Engl J Med. 2003; 349(23):2191–2200. [PubMed: 14657426]
46. Summers RM. Improving the accuracy of CT colonography interpretation: computer-aided diagnosis. Gastrointest Endosc Clin N Am. 2010; 20:245–257. [PubMed: 20451814]
47. Wang S, Yao J, Petrick N, Summers RM. Combining statistical and geometric features for colonic polyp detection in CTC based on multiple kernel learning. Int J Comput Intell Appl. 2010; 9(1):1–15. [PubMed: 20953299]
48. Suzuki K, Zhang J, Xu JW. Massive-training artificial neural network coupled with Laplacian-eigenfunction-based dimensionality reduction for computer-aided detection of polyps in CT colonography. IEEE Trans Med Imag. 2010; 29(11):1907–1917.
49. Wang S, McKenna MT, Nguyen TB, Burns JE, Petrick N, Sahiner B, Summers RM. Seeing is believing: video classification for computed tomographic colonography using multiple-instance learning. IEEE Trans Med Imag. 2012; 31(5):1141–1153.
50. Franaszek M, Summers RM, Pickhardt PJ, Choi JR. Hybrid segmentation of colon filled with air and opacified fluid for CT colonography. IEEE Trans Med Imag. 2006; 25(3):358–368.

51. Jerebko AK, Malley JD, Franaszek M, Summers RM. Support vector machines committee classification method for computer-aided polyp detection in CT colonography. Acad Radiol. 2005; 12(4):479–486. [PubMed: 15831422]

52. Ng, AY. Feature selection, L1 vs. L2 regularization, and rotational invariance. the 21st International Conference on Machine Learning; 2004.

53. Zhu J, Rosset S, Hastie T, Tibshirani R. 1-norm support vector machines. Advances in Neural Information Processing Systems. 2004; 16

54. Boyd, S.; Vandenberghe, L. Convex Optimization. Cambridge University Press; Cambridge, England: 2004.

Biography

Dr. Shijun Wang received his PhD degree in Control Science and Engineering from Tsinghua University, China, where his research focused on machine learning and complex systems. He earned a BS in Electronic Engineering at Beihang University and an MS in Communication at Second Aerospace Science Academy, China. Dr. Shijun Wang's current research interests in the Imaging Biomarkers and Computer-Aided Diagnosis Laboratory include machine learning, statistical image analysis and their applications in computer-aided diagnosis. He is an associate editor of Medical Physics and reviewer for IEEE TMI, IEEE TBME, Journal of Artificial Intelligence Research, Journal of Magnetic Resonance Imaging, Medical Physics, Pattern Analysis & Applications, and Journal of Theoretical Biology.

Appendix A. Proof of Theorem 1

Proof

By using the Lagrange multipliers optimization method [54], we transfer the constrained optimization problem 3 into the following unconstrained primal Lagrange function:

$$\begin{aligned}
&L_p^1(\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\chi},\boldsymbol{\gamma},\boldsymbol{\kappa},\boldsymbol{\lambda})\\
&=\tfrac{1}{2}\|\boldsymbol{w}\|^2+\tfrac{C_1}{2}\sum_{i=1}^{p}\sum_{j=1}^{q}\xi_{ij}+\tfrac{C_2}{2}\sum_{m=1}^{r}\left(\sum_{j=1}^{q}\eta_{mj}+\sum_{i=1}^{p}\mu_{mi}\right)\\
&-\sum_{i=1}^{p}\sum_{j=1}^{q}\alpha_{ij}((\langle\boldsymbol{w},\phi(x_i^+)\rangle-\langle\boldsymbol{w},\phi(x_j^-)\rangle)+\xi_{ij}-1)-\sum_{i=1}^{p}\sum_{j=1}^{q}\beta_{ij}\xi_{ij}\\
&-\sum_{m=1}^{r}\sum_{j=1}^{q}\chi_{mj}((\langle\boldsymbol{w},\phi(x_m)\rangle-\langle\boldsymbol{w},\phi(x_j^-)\rangle)\\
&+M(1-d_m)+\eta_{mj}-1)-\sum_{m=1}^{r}\sum_{j=1}^{q}\gamma_{mj}\eta_{mj}\\
&-\sum_{m=1}^{r}\sum_{i=1}^{p}\kappa_{mi}(-1\times(\langle\boldsymbol{w},\phi(x_m)\rangle-\langle\boldsymbol{w},\phi(x_i^+)\rangle)+Md_m+\mu_{mi}-1)\\
&-\sum_{m=1}^{r}\sum_{i=1}^{p}\lambda_{mi}\mu_{mi}.
\end{aligned}$$

The Karush–Kuhn–Tucker (KKT) conditions [54] for optimal primal variables $\boldsymbol{w}$, $\boldsymbol{\xi}$, $\boldsymbol{\eta}$, and $\boldsymbol{\mu}$ are

Stationarity:

$$\begin{gathered}\frac{\partial L_p^1}{\partial \boldsymbol{w}}=\boldsymbol{w}-\sum_{i=1}^{p}\sum_{j=1}^{q}\alpha_{ij}(\phi(x_i^+)-\phi(x_j^-))\\ -\sum_{m=1}^{r}\sum_{j=1}^{q}\chi_{mj}(\phi(x_m)-\phi(x_j^-))\\ -\sum_{m=1}^{r}\sum_{i=1}^{p}\kappa_{mi}(-1\times(\phi(x_m)-\phi(x_i^+)))=0,\\ \frac{\partial L_p^1}{\partial \boldsymbol{\xi}}=\frac{C_1}{2}\boldsymbol{e}-\boldsymbol{\alpha}-\boldsymbol{\beta}=0\\ \frac{\partial L_p^1}{\partial \boldsymbol{\eta}}=\frac{C_2}{2}\boldsymbol{e}-\boldsymbol{\chi}-\boldsymbol{\gamma}=0,\\ \frac{\partial L_p^1}{\partial \boldsymbol{\mu}}=\frac{C_2}{2}\boldsymbol{e}-\boldsymbol{\kappa}-\boldsymbol{\lambda}=0;\end{gathered}$$

Primal feasibility:

$$\begin{gathered}(\langle \boldsymbol{w},\phi(x_i^+)\rangle-\langle \boldsymbol{w},\phi(x_j^-)\rangle)\geq 1-\xi_{ij},\quad \xi_{ij}\geq 0,\quad i=1,2,\ldots,p,j=1,2,\ldots,q,\\ (\langle \boldsymbol{w},\phi(x_m)\rangle-\langle \boldsymbol{w},\phi(x_j^-)\rangle)+M(1-d_m)\geq 1-\eta_{mj}\quad \eta_{mj}\geq 0,\quad m=1,2,\ldots,r,j=1,2,\ldots,q,\\ -1\times(\langle \boldsymbol{w},\phi(x_m)\rangle-\langle \boldsymbol{w},\phi(x_i^+)\rangle)+Md_m\geq 1-\mu_{mi},\quad \mu_{mi}\geq 0,\quad m=1,2,\ldots,r,i=1,2,\ldots,p;\end{gathered}$$

Dual feasibility:

$$\begin{gathered}\alpha_{ij}\geq 0,\quad i=1,2,\ldots,p,\ j=1,2,\ldots,q,\\ \beta_{ij}\geq 0,\quad i=1,2,\ldots,p,\ j=1,2,\ldots,q,\\ \chi_{mj}\geq 0,\quad m=1,2,\ldots,r,\ j=1,2,\ldots,q,\\ \gamma_{mj}\geq 0,\quad m=1,2,\ldots,r,\ j=1,2,\ldots,q,\\ \kappa_{mi}\geq 0,\quad m=1,2,\ldots,r,\ i=1,2,\ldots,p,\\ \lambda_{mi}\geq 0,\quad m=1,2,\ldots,r,\ i=1,2,\ldots,p;\end{gathered}$$

Complementary slackness:

$$\begin{gathered}\alpha_{ij}((\langle \boldsymbol{w},\phi(x_i^+)\rangle-\langle \boldsymbol{w},\phi(x_j^-)\rangle)+\xi_{ij}-1)=0,\\ \beta_{ij}\xi_{ij}=0,\ i=1,2,\ldots,p,j=1,2,\ldots,q,\\ \chi_{mj}((\langle \boldsymbol{w},\phi(x_m)\rangle-\langle \boldsymbol{w},\phi(x_j^-)\rangle)+M(1-d_m)+\eta_{mj}-1)=0,\\ \gamma_{mj}\eta_{mj}=0,\ m=1,2,\ldots,r,j=1,2,\ldots,q,\\ \kappa_{mi}(-1\times(\langle \boldsymbol{w},\phi(x_m)\rangle-\langle \boldsymbol{w},\phi(x_i^+)\rangle)+Md_m+\mu_{mi}-1)=0,\\ \lambda_{mi}\mu_{mi}=0,m=1,2,\ldots,r,i=1,2,\ldots,p,\end{gathered}$$

where

$$\begin{gathered}\boldsymbol{\alpha}=\{\alpha_{11},\alpha_{12},\ldots,\alpha_{1q},\alpha_{21},\alpha_{22}\ldots,\alpha_{2q},\ldots,\alpha_{p1}\ldots,\alpha_{pq}\},\\ \boldsymbol{\beta}=\{\beta_{11},\beta_{12}\ldots,\beta_{1q},\beta_{21},\beta_{22}\ldots,\beta_{2q},\ldots,\beta_{p1}\ldots,\beta_{pq}\},\\ \boldsymbol{\chi}=\{\chi_{11},\chi_{12}\cdots,\chi_{1q},\chi_{21},\chi_{22}\cdots,\chi_{2q},\cdots,\chi_{r1}\cdots,\chi_{rq}\},\\ \boldsymbol{\gamma}=\{\gamma_{11},\gamma_{12}\cdots,\gamma_{1q},\gamma_{21},\gamma_{22}\cdots,\gamma_{2q},\cdots,\gamma_{r1}\cdots,\gamma_{rq}\},\\ \boldsymbol{\kappa}=\{\kappa_{11},\kappa_{12}\ldots,\kappa_{1p},\kappa_{21},\kappa_{22}\ldots,\kappa_{2p},\ldots,\kappa_{r1}\ldots,\kappa_{rp}\},\\ \boldsymbol{\lambda}=\{\lambda_{11},\lambda_{12}\ldots,\lambda_{1p},\lambda_{21},\lambda_{22}\ldots,\lambda_{2p},\ldots,\lambda_{r1}\ldots,\lambda_{r_p}\},\end{gathered}$$

, $\boldsymbol{e}$ is a vector filled with all ones.

The optimal $\boldsymbol{w}$ can be achieved at:

$$\boldsymbol{w}=\sum_{i=1}^{p}\sum_{j=1}^{q}\alpha_{ij}(\phi(x_i^+)-\phi(x_j^-))+\sum_{m=1}^{r}\sum_{j=1}^{q}\chi_{mj}(\phi(x_m)-\phi(x_j^-))+\sum_{m=1}^{r}\sum_{i=1}^{p}\kappa_{mi}(-1\times(\phi(x_m)-\phi(x_i^+))).$$

$$\begin{aligned}
K^{PNPN}_{i_1j_1,i_2j_2}&\\
&=k_{i_1i_2}-k_{i_1j_2}-k_{j_1i_2}+k_{j_1j_2},\ K^{PNUN}_{ij_1,mj_2}\\
&=k_{im}-k_{ij_2}-k_{j_1m}+k_{j_1j_2},\ K^{PNUP}_{i_1j,mi_2}\\
&=k_{i_1m}-k_{i_1i_2}-k_{jm}+k_{ji_2},\ K^{UNUN}_{m_1j_1,m_2j_2}\\
&=k_{m_1m_2}-k_{m_1j_2}-k_{j_1m_2}+k_{j_1j_2},\ K^{UNUP}_{m_1j,m_2i}\\
&=k_{m_1m_2}-k_{m_1i}-k_{jm_2}+k_{ji},\ K^{UPUP}_{m_1i_1,m_2i_2}\\
&=k_{m_1m_2}-k_{m_1i_2}-k_{i_1m_2}+k_{i_1i_2}
\end{aligned}$$

Let us define: , $k_{ij} = \langle \varphi(x_i), \varphi(x_j) \rangle$, x_i and x_j are samples from the set denoted by the corresponding superscripts.

In the above definitions, *P* means positive training samples; *N* means negative training samples; *U* means unknown test samples. $\boldsymbol{K}^{PNPN}$ defines the kernel matrix between positive–negative training sample pairs; $\boldsymbol{K}^{PNUN}$ defines the kernel matrix between positive–negative training sample pairs and test-negative sample pairs; $\boldsymbol{K}^{PNUP}$ defines the kernel matrix between positive–negative training sample pairs and test-positive sample pairs; $\boldsymbol{K}^{UNUN}$ defines the kernel matrix between test-negative sample pairs; $\boldsymbol{K}^{UNUP}$ defines the kernel matrix between test-negative sample pairs and test-positive sample pairs; $\boldsymbol{K}^{UPUP}$ defines the kernel matrix between test-positive sample pairs. Here negative and positive samples are only from training set and test samples are only from test set.

Therefore:

$$\begin{aligned}
&L_p^1(\boldsymbol{w},\boldsymbol{\xi},\boldsymbol{\eta},\boldsymbol{\mu},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\chi},\boldsymbol{\gamma},\boldsymbol{\kappa},\boldsymbol{\lambda})\\
=&-\tfrac{1}{2}(\boldsymbol{\alpha}^T\boldsymbol{K}^{PNPN}\boldsymbol{\alpha}+2\boldsymbol{\alpha}^T\boldsymbol{K}^{PNUN}\boldsymbol{\chi}-2\boldsymbol{\alpha}^T\boldsymbol{K}^{PNUP}\boldsymbol{\kappa}+\boldsymbol{\chi}^T\boldsymbol{K}^{UNUN}\boldsymbol{\chi}\\
&-2\boldsymbol{\chi}^T\boldsymbol{K}^{UNUP}\boldsymbol{\kappa}+\boldsymbol{\kappa}^T\boldsymbol{K}^{UPUP}\boldsymbol{\kappa})+\sum_{i=1}^{p}\sum_{j=1}^{q}\alpha_{ij}\\
&-\sum_{m=1}^{r}\sum_{j=1}^{q}\chi_{mj}(M(1-d_m)-1)\\
&-\sum_{m=1}^{r}\sum_{i=1}^{p}\kappa_{mi}(Md_m-1).
\end{aligned}$$

After removing primal variables, we get dual representation of the optimization problem as follows:

$$\begin{aligned}
\max L_p^1(\boldsymbol{\alpha},\boldsymbol{\chi},\boldsymbol{\kappa})=&-\tfrac{1}{2}\times(\boldsymbol{\alpha}^T\boldsymbol{K}^{PNPN}\boldsymbol{\alpha}+2\boldsymbol{\alpha}^T\boldsymbol{K}^{PNUN}\boldsymbol{\chi}-2\boldsymbol{\alpha}^T\boldsymbol{K}^{PNUP}\boldsymbol{\kappa}\\
&+\boldsymbol{\chi}^T\boldsymbol{K}^{UNUN}\boldsymbol{\chi}-2\boldsymbol{\chi}^T\boldsymbol{K}^{UNUP}\boldsymbol{\kappa}+\boldsymbol{\kappa}^T\boldsymbol{K}^{UPUP}\boldsymbol{\kappa})\\
&+\boldsymbol{\alpha}^T\boldsymbol{e}-\sum_{m=1}^{r}\sum_{j=1}^{q}\chi_{kj}(M(1-d_m)-1)-\sum_{m=1}^{r}\sum_{i=1}^{p}\kappa_{mi}(Md_m-1),
\end{aligned}$$

with constraints: $0 \le \boldsymbol{\alpha} \le C_1/2$, $0 \le \boldsymbol{\chi} C_2/2$, $0 \le \boldsymbol{\kappa} C_2/2$. Thus, the Lagrangian of the maximization problem can be defined as

$$\begin{aligned} &L_p^1(\boldsymbol{\alpha},\boldsymbol{\chi},\boldsymbol{\kappa},\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{\theta},\boldsymbol{\rho},\boldsymbol{\sigma},\boldsymbol{\tau}) \\ =&-\tfrac{1}{2}\times(\boldsymbol{\alpha}^T\boldsymbol{K}^{PNPN}\boldsymbol{\alpha}+2\boldsymbol{\alpha}^T\boldsymbol{K}^{PNUN}\boldsymbol{\chi}-2\boldsymbol{\alpha}^T\boldsymbol{K}^{PNUP}\boldsymbol{\kappa} \\ &+\boldsymbol{\chi}^T\boldsymbol{K}^{UNUN}\boldsymbol{\chi}-2\boldsymbol{\chi}^T\boldsymbol{K}^{UNUP}\boldsymbol{\kappa}+\boldsymbol{\kappa}^T\boldsymbol{K}^{UPUP}\boldsymbol{\kappa}) \\ &+\boldsymbol{\alpha}^T\boldsymbol{e}-\boldsymbol{\chi}^T\boldsymbol{d}^N-\boldsymbol{\kappa}^T\boldsymbol{d}^p+\boldsymbol{\nu}^T\left(\tfrac{C_1}{2}\boldsymbol{e}-\boldsymbol{\alpha}\right)+\boldsymbol{o}^T\boldsymbol{\alpha} \\ &+\boldsymbol{\theta}^T\left(\tfrac{C_2}{2}\boldsymbol{e}-\chi\right)+\boldsymbol{\rho}^T\boldsymbol{\chi}+\boldsymbol{\sigma}^T\left(\tfrac{C_2}{2}\boldsymbol{e}-\boldsymbol{\kappa}\right)+\boldsymbol{\tau}^T\boldsymbol{\kappa}, \end{aligned}$$

where $d^N_{(m-1)*q+j}=(M(1-d_m)-1)$, $m = 1, 2, \ldots, r$, $j = 1, 2, \ldots, q$, $d^P_{(m-1)*p+i}=(Md_m-1)$, $m = 1, 2, \ldots, r$, $i = 1, 2, \ldots, p$, s.t. $\boldsymbol{\nu}\geq 0$, $\boldsymbol{o}\geq 0$, $\boldsymbol{\theta}\geq 0$, $\rho\geq 0$, $\sigma\geq 0$, $\tau\geq 0$.

Let us define

$$\boldsymbol{\omega}=\begin{pmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\chi}\\ \boldsymbol{\kappa}\end{pmatrix},\quad \boldsymbol{\psi}=\begin{pmatrix}\boldsymbol{\nu}\\ \boldsymbol{\theta}\\ \boldsymbol{\sigma}\end{pmatrix},\quad \boldsymbol{\zeta}=\begin{pmatrix}\boldsymbol{o}\\ \boldsymbol{\rho}\\ \boldsymbol{\tau}\end{pmatrix}\quad \text{and}$$
$$K=\begin{bmatrix}+\boldsymbol{K}^{PNPN}+\boldsymbol{K}^{PNUN}-\boldsymbol{K}^{PNUP}\\ +\boldsymbol{K}^{UNPN}+\boldsymbol{K}^{UNUN}-\boldsymbol{K}^{UNUP}\\ -\boldsymbol{K}^{UPPN}-\boldsymbol{K}^{UPUN}+\boldsymbol{K}^{UPUP}\end{bmatrix},$$

where $\boldsymbol{K}^{PNUP}=\boldsymbol{K}^{UPPN^T}$, $\boldsymbol{K}^{PNUN}=\boldsymbol{K}^{UNPN^T}$, $\boldsymbol{K}^{UNUP}=\boldsymbol{K}^{UPUN^T}$.

$$\begin{aligned} &L_p^1(\boldsymbol{\alpha},\boldsymbol{\chi},\boldsymbol{\kappa},\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{\theta},\boldsymbol{\rho},\boldsymbol{\sigma},\boldsymbol{\tau})=L_p^1(\boldsymbol{\omega},\boldsymbol{\psi},\boldsymbol{\zeta}) \\ =&-\tfrac{1}{2}\boldsymbol{\omega}^T\boldsymbol{K}\boldsymbol{\omega}+\boldsymbol{\omega}^T\left(\begin{pmatrix}\boldsymbol{e}\\ -\boldsymbol{d}^N\\ -\boldsymbol{d}^P\end{pmatrix}-\boldsymbol{\psi}+\boldsymbol{\zeta}\right)+\boldsymbol{\psi}^T\begin{pmatrix}\frac{C_1}{2}\boldsymbol{e}\\ \frac{C_2}{2}\boldsymbol{e}\\ \frac{C_2}{2}\boldsymbol{e}\end{pmatrix}. \end{aligned}$$

Let us define

$$\boldsymbol{d_3}=\begin{pmatrix}\boldsymbol{e}\\ -\boldsymbol{d}^N\\ -\boldsymbol{d}^P\end{pmatrix}\quad \text{and}\quad \boldsymbol{e_3}=\begin{pmatrix}\frac{C_1}{2}\boldsymbol{e}\\ \frac{C_2}{2}\boldsymbol{e}\\ \frac{C_2}{2}\boldsymbol{e}\end{pmatrix}.$$

Therefore

$$\begin{aligned} &L_p^1(\boldsymbol{\omega},\boldsymbol{\psi},\boldsymbol{\zeta})=-\tfrac{1}{2}\boldsymbol{\omega}^T\boldsymbol{K}\boldsymbol{\omega}+\boldsymbol{\omega}^T(\boldsymbol{d}_3-\boldsymbol{\psi}+\boldsymbol{\zeta})+\boldsymbol{\psi}^T\boldsymbol{e_3}, \\ &\text{s.t.}\,\boldsymbol{\omega}\geq 0,\boldsymbol{\psi}\geq 0,\boldsymbol{\zeta}\geq 0. \end{aligned}$$

Based on duality, we have the following equivalent problems:

$$\max_{\boldsymbol{\omega}\geq 0\boldsymbol{\psi}\geq 0,\boldsymbol{\zeta}\geq 0}\min L_p^1(\boldsymbol{\omega},\boldsymbol{\psi},\boldsymbol{\zeta})=\min_{\boldsymbol{\psi}\geq 0,\boldsymbol{\zeta}\geq 0\boldsymbol{\omega}\geq 0}\max L_p^1(\boldsymbol{\omega},\boldsymbol{\psi},\boldsymbol{\zeta}).$$

The inner maximization could be achieved at:

$$\frac{\partial L_p^1(\boldsymbol{\omega},\boldsymbol{\psi},\boldsymbol{\zeta})}{\partial \boldsymbol{\omega}} = -\boldsymbol{K}\boldsymbol{\omega} + (\boldsymbol{d_3} - \boldsymbol{\psi} + \boldsymbol{\zeta}) = 0$$
$$\Rightarrow \boldsymbol{\omega} = \boldsymbol{K}^{-1}(\boldsymbol{d}_3 - \boldsymbol{\psi} + \boldsymbol{\zeta}).$$

So

$$\min_{\boldsymbol{\psi}\geq 0,\boldsymbol{\zeta}\geq 0}\max_{\boldsymbol{\omega}\geq 0} L_p^1(\boldsymbol{\omega},\boldsymbol{\psi},\boldsymbol{\zeta}) = \min_{\boldsymbol{\psi}\geq 0,\boldsymbol{\zeta}\geq 0} -\frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{K}\boldsymbol{\omega}$$
$$+\boldsymbol{\omega}^T(\boldsymbol{d_3} - \boldsymbol{\psi} + \boldsymbol{\zeta}) + \boldsymbol{\psi}^T\boldsymbol{e_3} \big| \boldsymbol{\omega} = \boldsymbol{K}^{-1}(\boldsymbol{d_3} - \boldsymbol{\psi} + \boldsymbol{\zeta}).$$
$$= \min_{\boldsymbol{\psi}\geq 0,\boldsymbol{\zeta}\geq 0} \frac{1}{2}(\boldsymbol{d_3} - \boldsymbol{\psi} + \boldsymbol{\zeta})^T\boldsymbol{K}^{-1}(\boldsymbol{d_3} - \boldsymbol{\psi} + \boldsymbol{\zeta}) + \boldsymbol{\psi}^T\boldsymbol{e_3}.$$

Let $t \geq 0$ be the upper limit of the minimization problem:

$$t \geq \min_{\boldsymbol{\psi}\geq 0,\boldsymbol{\zeta}\geq 0} \frac{1}{2}(\boldsymbol{d_3} - \boldsymbol{\psi} + \boldsymbol{\zeta})^T\boldsymbol{K}^{-1}(\boldsymbol{d_3} - \boldsymbol{\psi} + \boldsymbol{\zeta}) + \boldsymbol{\psi}^T\boldsymbol{e_3}.$$

Using the Schur complement [54], we will get

$$\begin{bmatrix} \boldsymbol{K} & \frac{(\boldsymbol{d_3} - \boldsymbol{\psi} + \boldsymbol{\zeta})}{\sqrt{2}} \\ \frac{(\boldsymbol{d_3} - \boldsymbol{\psi} + \boldsymbol{\zeta})^T}{\sqrt{2}} & t - \boldsymbol{\psi}^T\boldsymbol{e_3} \end{bmatrix} \geq 0,$$
$$\boldsymbol{\psi} \geq 0, \boldsymbol{\zeta} \geq 0.$$

So we have the following SDP problem:

$$\min_{\boldsymbol{d_3},\boldsymbol{\psi},\boldsymbol{\zeta}} t$$
$$\text{s.t.}$$
$$\begin{bmatrix} \boldsymbol{K} & \frac{(\boldsymbol{d_3} - \boldsymbol{\psi} + \boldsymbol{\zeta})}{\sqrt{2}} \\ \frac{(\boldsymbol{d_3} - \boldsymbol{\psi} + \boldsymbol{\zeta})^T}{\sqrt{2}} & t - \boldsymbol{\psi}^T\boldsymbol{e_3} \end{bmatrix} \geq 0,$$
$$\boldsymbol{\psi} \geq 0, \boldsymbol{\zeta} \geq 0.$$

In practice, we found that adding a regularizer *diag*($\boldsymbol{I_1}/C_1$, $\boldsymbol{I_2}/C_2$, $\boldsymbol{I_3}/C_2$) to $\boldsymbol{K}$ will increase the positive definiteness of $\boldsymbol{K}$ and lead to better performance, where *diag* is a diagonal matrix and $\boldsymbol{I_1}$, $\boldsymbol{I_2}$, and $\boldsymbol{I_3}$ are identify matrices having the same size as $\boldsymbol{K}^{PNPN}$, $\boldsymbol{K}^{UNUN}$, and $\boldsymbol{K}^{UPUP}$, respectively.

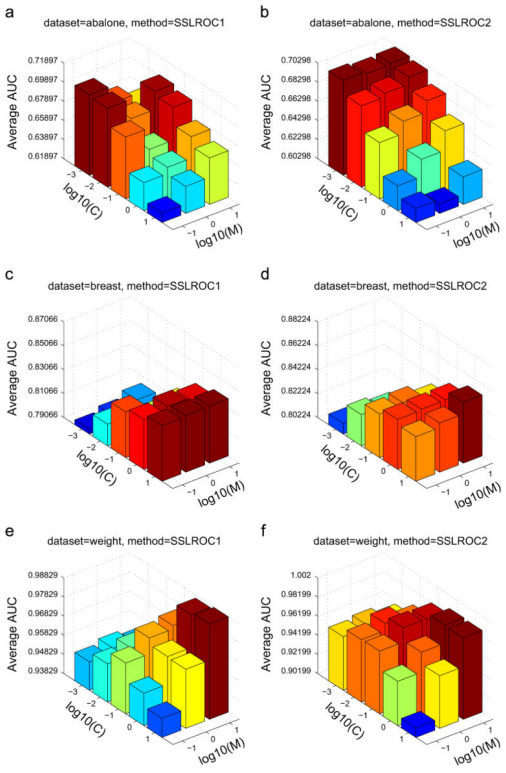

Fig. 1.

Average AUCs of 5 × 2-fold CV on three example UCI datasets. The first column corresponds to SSLROC1 and the second column corresponds to SSLROC2. Each row corresponds to one dataset. For both SSLROC1 and SSLROC2, we show results in the same parameter space spanned by log 10(*C*) and log 10(*M*).

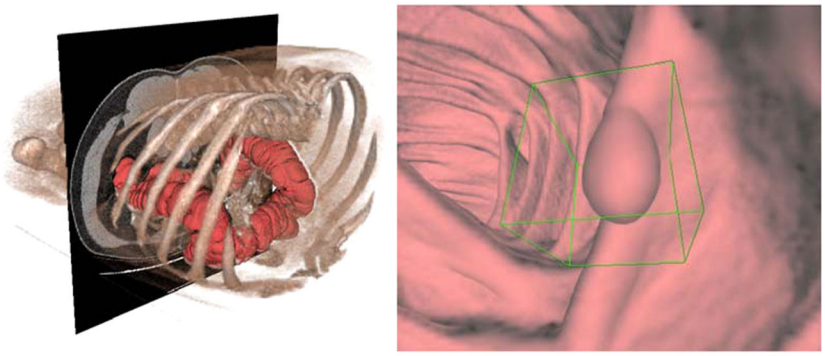

Fig. 2.

3D volume rendering of a segmented colon (left figure) with spine and ribs; a typical colonic polyp on the fold (right figure).

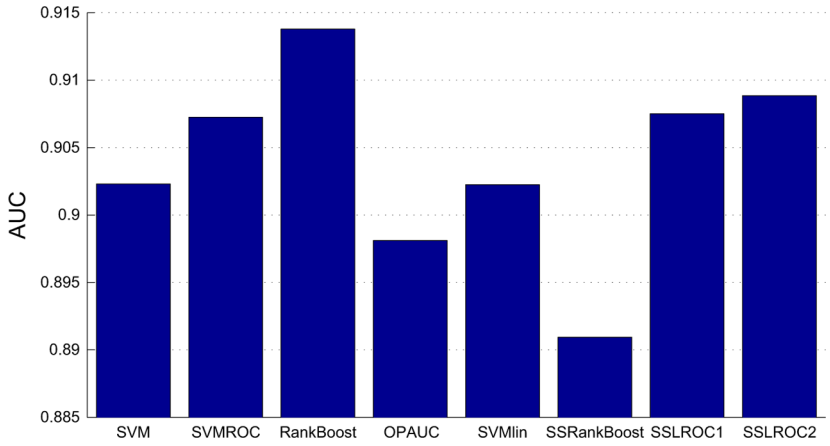

Fig. 3.
Comparison of AUCs of eight methods on the CTC dataset.

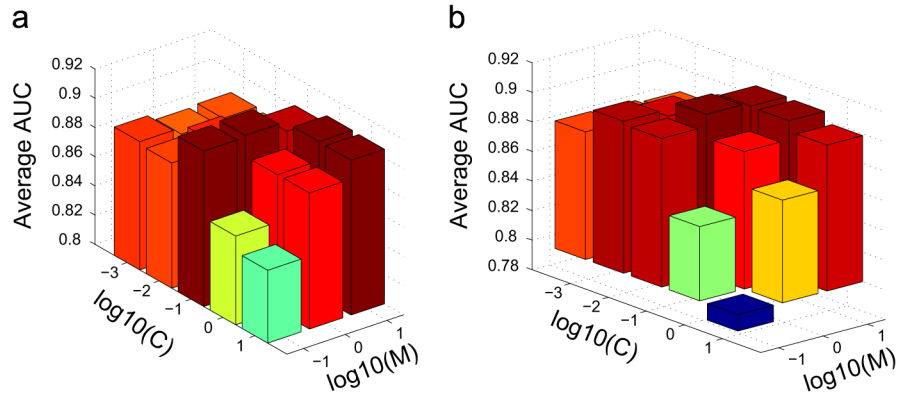

Fig. 4.

Average AUC of SSLROC1 (a) and SSLROC2 (b) on the CTC dataset when different *C* values (classifier complexity and training error trade-off parameter) and *M* values (margin size parameter) were used in the experiment. (a) SSLROC1 and (b) SSLROC2.

Table 1

Characteristics of the 34 UCI datasets employed in this study. Under the class labels, “rest” designates that it was a multi-class problem and that the rest of the classes were combined into one class.

Dataset	Name	#	Attr	Class label (+1)	Class label (−1)
1	Abalone	4177	8	Female	Male, infant
2	Arcene_train	100	10,000	Positive	Negative
3	Blood	748	5	Donated blood	Did not donate
4	Breast	106	9	Car, fad, mas	gla, con, adi
5	Bupaliver	345	6	>5 drinks	<5 drinks
6	Cancer_wbc	669	9	Malignant	Benign
7	Cardio	74	10	Alive after 1 year	Died before 1 year
8	cmc	1473	9	Long/use of contraceptives	No contraceptive use
9	cnae_9	1080	856	Category: range 1–5	Category: range 6–9
10	Credit_g	1000	20	Good credit	Bad credit
11	Derm	366	34	4,5,6	1,2,3
12	*E.coli*	336	6	pp	rest
13	Glass	214	9	7th type	Rest
14	Heart	270	14	Absence	Presence
15	Hepatitis	155	19	Die	Live
16	House	435	16	Democrat	Republican
17	Ionosphere	351	34	Bad	Good
18	Iris	150	4	setosa	Versicolor, virginica
19	Kidney_inflam	120	6	Bladder inflammation	No inflammation
20	Kr vs. kp	3196	36	White wins	White loses
21	Mushroom	8124	21	Edible	Poisonous
22	Parkinsons	197	23	Parkinson’s	Healthy
23	pima	768	8	Positive for diabetes	No diabetes
24	Post_op	90	8	Patient discharged (s)	Rest
25	sonar	208	60	Rock	Mine
26	Spectf	267	45	1	0
27	Statlog	690	14	Credit approved	Not approved

Dataset	Name	#	Attr	Class label (+1)	Class label (−1)
28	Survival	306	3	Survived 5+ years	Died within 5 years
29	Teach	151	5	Low	Medium, high
30	Tictactoe	958	9	x wins	x loses
31	Vehicle	846	18	van, bus	saab, opel
32	Weight	625	4	Right-leaning	Balanced/left-leaning
33	Wine	178	12	Cultivar 3	Cultivar 1 and 2
34	Zoo	101	17	Aquatic animals	Not aquatic

Table 2

Average AUCs of eight methods on the 34 UCI machine learning datasets.

Dataset	SVM		SVMROC		RankBoost		OPAUC		SVMlin		SSRankBoost		SSLROC1		SSLROC2	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
1	0.700	0.078	0.699	0.079	0.620	0.083	0.701	0.075	0.704	0.078	0.654	0.080	0.703	0.077	0.702	0.078
2	0.806	0.092	0.806	0.092	0.659	0.086	0.701	0.126	0.811	0.094	0.707	0.083	0.815	0.072	0.818	0.069
3	0.735	0.039	0.735	0.036	0.689	0.072	0.716	0.056	0.725	0.045	0.691	0.077	0.737	0.042	0.736	0.048
4	0.906	0.026	0.927	0.038	0.946	0.015	0.877	0.030	0.926	0.031	0.915	0.026	0.910	0.038	0.930	0.026
5	0.631	0.086	0.627	0.088	0.631	0.059	0.641	0.082	0.648	0.095	0.652	0.088	0.638	0.081	0.632	0.084
6	0.993	0.006	0.993	0.006	0.974	0.030	0.993	0.006	0.994	0.006	0.984	0.013	0.994	0.005	0.993	0.005
7	0.984	0.013	0.979	0.013	0.963	0.045	0.984	0.015	0.982	0.013	0.978	0.016	0.983	0.013	0.979	0.012
8	1.00	0.001	1.00	0.001	1.00	0.000	1.00	0.001	1.00	0.001	1.00	0.000	0.999	0.001	1.00	0.001
9	0.932	0.046	0.938	0.049	0.904	0.056	0.958	0.021	0.959	0.025	0.927	0.043	0.943	0.043	0.943	0.043
10	0.734	0.092	0.734	0.092	0.700	0.096	0.725	0.088	0.719	0.099	0.699	0.087	0.733	0.092	0.734	0.091
11	0.993	0.008	0.994	0.007	0.960	0.032	0.994	0.007	0.991	0.011	0.981	0.012	0.996	0.005	0.996	0.005
12	0.948	0.037	0.952	0.044	0.860	0.112	0.931	0.052	0.917	0.070	0.922	0.069	0.947	0.050	0.951	0.047
13	0.963	0.044	0.964	0.045	0.905	0.110	0.954	0.061	0.964	0.034	0.934	0.063	0.970	0.030	0.970	0.036
14	0.923	0.039	0.925	0.036	0.880	0.051	0.928	0.038	0.923	0.038	0.892	0.050	0.925	0.037	0.925	0.038
15	0.856	0.061	0.853	0.061	0.753	0.080	0.842	0.056	0.854	0.052	0.803	0.065	0.854	0.057	0.851	0.060
16	0.988	0.012	0.988	0.011	0.983	0.016	0.989	0.010	0.989	0.010	0.984	0.016	0.991	0.006	0.990	0.009
17	0.951	0.037	0.946	0.044	0.914	0.049	0.889	0.049	0.933	0.023	0.901	0.059	0.964	0.026	0.962	0.028
18	1.00	0.000	1.00	0.000	0.998	0.006	1.00	0.000	1.00	0.000	0.999	0.002	1.00	0.000	1.00	0.000
19	1.00	0.000	1.00	0.000	1.00	0.000	1.00	0.000	1.00	0.000	1.00	0.000	1.00	0.000	1.00	0.000
20	0.884	0.035	0.884	0.046	0.934	0.042	0.873	0.045	0.891	0.049	0.934	0.042	0.892	0.033	0.888	0.041
21	0.954	0.037	0.957	0.036	0.955	0.040	0.954	0.039	0.949	0.041	0.956	0.035	0.961	0.023	0.961	0.023
22	0.878	0.040	0.878	0.041	0.885	0.047	0.879	0.043	0.877	0.037	0.885	0.039	0.878	0.040	0.879	0.041
23	0.766	0.066	0.767	0.066	0.729	0.060	0.766	0.076	0.778	0.045	0.736	0.043	0.784	0.054	0.777	0.049
24	0.604	0.070	0.606	0.072	0.593	0.059	0.607	0.069	0.595	0.082	0.606	0.062	0.607	0.081	0.606	0.065
25	0.830	0.059	0.850	0.045	0.842	0.048	0.839	0.045	0.833	0.043	0.837	0.055	0.845	0.039	0.849	0.047
26	0.852	0.064	0.852	0.064	0.736	0.101	0.852	0.068	0.834	0.081	0.783	0.058	0.849	0.063	0.852	0.065
27	0.928	0.032	0.925	0.027	0.887	0.036	0.924	0.030	0.926	0.034	0.885	0.046	0.928	0.028	0.927	0.031

Dataset	SVM		SVMROC		RankBoost		OPAUC		SVMlin		SSRankBoost		SSLROC1		SSLROC2	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std	Avg	Std
28	0.646	0.102	0.644	0.101	0.616	0.067	0.646	0.086	0.642	0.106	0.616	0.077	0.647	0.069	0.658	0.084
29	0.708	0.074	0.710	0.086	0.684	0.062	0.670	0.079	0.675	0.093	0.700	0.068	0.731	0.053	0.726	0.049
30	0.750	0.105	0.744	0.110	0.762	0.091	0.679	0.070	0.692	0.083	0.765	0.089	0.774	0.074	0.782	0.064
31	0.949	0.033	0.951	0.036	0.859	0.050	0.850	0.074	0.937	0.041	0.877	0.034	0.956	0.029	0.966	0.024
32	0.988	0.009	0.989	0.009	0.958	0.028	0.974	0.013	0.988	0.010	0.913	0.034	0.988	0.013	0.988	0.013
33	1.00	0.001	1.00	0.001	0.976	0.053	1.00	0.001	1.00	0.001	0.994	0.008	1.00	0.000	1.00	0.000
34	0.989	0.010	0.990	0.009	0.992	0.007	0.990	0.010	0.991	0.011	0.991	0.009	0.991	0.008	0.991	0.008
Avg	0.875	0.043	0.877	0.044	0.845	0.053	0.863	0.045	0.872	0.044	0.856	0.045	0.880	0.038	0.881	0.038

Table 3

Numbers of wins–ties–losses (superior–equal–inferior AUC) between the eight methods (pairwise) on the 34 UCI datasets. For three numbers shown in each entry of the table, the first is the number of wins of corresponding method shown on the left column compared with the corresponding method shown on the top; middle is the number of ties between them and the third is the number of losses. (M1:SVM; M2:SVMROC; M3:RankBoost; M4:OPAUC; M5:SVMlin; M6:SSRankBoost; M7:SSLROC1; M8:SSLROC2).

Method	M1	M2	M3	M4	M5	M6	M7	M8
M1	0–34–0	13–6–15	25–1–8	17–2–15	19–2–13	23–1–10	7–2–25	5–2–27
M2	15–6–13	0–34–0	26–1–7	19–2–13	18–2–14	27–1–6	7–2–25	6–2–26
M3	8–1–25	7–1–26	0–34–0	11–1–22	9–1–24	9–3–22	5–1–28	5–1–28
M4	15–2–17	13–2–19	22–1–11	0–34–0	13–3–18	21–1–12	7–2–25	9–2–23
M5	13–2–19	14–2–18	24–1–9	18–3–13	0–34–0	23–1–10	7–2–25	9–2–23
M6	10–1–23	6–1–27	22–3–9	12–1–21	10–1–23	0–34–0	5–1–28	5–1–28
M7	25–2–7	25–2–7	28–1–5	25–2–7	25–2–7	28–1–5	0–34–0	15–6–13
M8	27–2–5	26–2–6	28–1–5	23–2–9	23–2–9	28–1–5	13–6–15	0–34–0

Table 4

p Values of the Wilcoxon signed rank tests between the four methods (pairwise). (M1:SVM; M2:SVMROC; M3:RankBoost; M4:OPAUC; M5:SVMlin; M6:SSRankBoost; M7:SSLROC1; M8:SSLROC2).

Method	M1	M2	M3	M4	M5	M6	M7	M8
M1	1.000	0.539	0.000	0.082	0.246	0.002	0.000	0.000
M2	0.539	1.000	0.000	0.026	0.145	0.000	0.003	0.000
M3	0.000	0.000	1.000	0.012	0.001	0.003	0.000	0.000
M4	0.082	0.026	0.012	1.000	0.117	0.098	0.001	0.001
M5	0.246	0.145	0.001	0.117	1.000	0.006	0.001	0.004
M6	0.002	0.000	0.003	0.098	0.006	1.000	0.000	0.000
M7	0.000	0.003	0.000	0.001	0.001	0.000	1.000	0.820
M8	0.000	0.000	0.000	0.001	0.004	0.000	0.820	1.000