**Noname manuscript No.**
(will be inserted by the editor)

# Least squares support vector machines with fast leave-one-out AUC optimization on imbalanced prostate cancer data

**Guanjin Wang · Jeremy Yuen-Chun Teoh · Jie Lu · Kup-Sze Choi**

the date of receipt and acceptance should be inserted later

**Abstract** Quite often, the available pre-biopsy data for early prostate cancer detection are imbalanced. When the least squares support vector machines (LS-SVMs) are applied to such scenarios, it becomes naturally desirable for us to introduce the well-known AUC performance index into the LS-SVMs framework to avoid bias towards majority classes. However, this may result in high computational complexity for the minimal leave-one-out error. In this paper, by introducing the parameter $\lambda$, a generalized Area under the ROC curve (AUC) performance index $R_{AUCLS}$ is developed to theoretically guarantee that $R_{AUCLS}$ linearly depends on the classical AUC performance index $R_{AUC}$. Based on both $R_{AUCLS}$ and the classical LS-SVM, a new AUC-based least squares support vector machine called AUC-LS-SVMs is proposed for directly and effectively classifying imbalanced prostate cancer data. The distinctive advantage of the proposed classifier AUC-LS-SVMs exists in that it can achieve the minimal leave-one-out error by quickly optimizing the parameter $\lambda$ in $R_{AUCLS}$ using the proposed fast leave-one-out cross validation (LOOCV) strategy. The proposed classifier is first evaluated using generic public datasets. Further experiments are then conducted on a real-world prostate cancer dataset to demonstrate the efficacy of our proposed classifier for early prostate cancer detection.

**Keywords** prostate cancer detection · imbalanced data · AUC performance index · least squares support vector machines · leave-one-out cross validation

✉Guanjin Wang
Discipline of Information Technology, Mathematics & Statistics, Murdoch University, Perth, Australia and Centre for Smart Health, School of Nursing, The Hong Kong Polytechnic University, Hong Kong, China
E-mail: Guanjin.Wang@murdoch.edu.au

Jeremy Yuen-Chun Teoh
Division of Urology, Department of Surgery, Prince of Wales Hospital, The Chinese University of Hong Kong, Hong Kong, China

Jie Lu
Centre for Artificial Intelligence, School of Software, Faculty of Engineering and Information Technology, University of Technology Sydney, Broadway, NSW, 2007, Australia

Kup-Sze Choi
Centre for Smart Health, School of Nursing, The Hong Kong Polytechnic University, Hong Kong, China

## 1 Introduction

Prostate cancer is one of the most common malignancies in males. In 2017, there were an estimated 161,360 new prostate cancer cases (nearly 10% of all the new cancer cases) and 26,730 deaths in the United States [1]. Traditionally, prostate cancer is screened by testing prostate-specific antigen (PSA) level in blood [6, 30], or performing digital rectal exam (DRE), where the doctor puts a gloved finger into the rectum to feel the prostate gland. If PSA level is elevated or DRE finding is abnormal, a higher risk of prostate cancer is suspected and prostate biopsy is recommended for further screening [10,11]. However, prostate biopsy as an invasive procedure may bring high risks of discomfort, infection and bleed to patients. Also, to avoid misdiagnosis, repeated biopsies are sometimes performed to guarantee the screening results, which may give patients emotional and physical pain. On the other hand, unnecessary biopsies should be avoided in patients with low-grade prostate cancer risks to prevent overdiagnosis and overtreatment. Therefore, to cut down unnecessary biopsies, researchers started to explore mathematical and computational methods to detect early prostate cancer using pre-biopsy information.

Support Vector Machines (SVMs) [17] as one of the most commonly used machine learning methods have been extensively taken to detect prostate cancer in the past decades. The main idea of SVMs is to construct a hyperplane in a high dimensional space, which has the largest margin between the closest training data points from different classes. The high dimensional space is mapped from the original finite dimensional space to make the separation easier in that space by using the kernel trick. The magic of kernel trick lies in that it is possible to compute the separating hyperplane without explicitly carrying out the mapping into the feature space, which reduces the computational cost. SVMs are very popular in the prostate cancer research field due to its ability to successfully generate high generalization performance. For example, Çınar et al. [15] designed a SVMs classifier based expert system for early diagnosis of prostate cancer by using prostate volume, density and other features to avoid biopsy. SVMs with polynomial based kernel function achieved the best performance (accuracy: 79%) among all the comparative methods, including artificial neural network. Li et al. [27] proposed a non-invasive prostate cancer screening methods using serum surface-enhanced Raman scattering (SERS) and SVMs via peripheral blood samples. The experimental results showed that the SVMs based diagnostic model achieved the accuracy of 98.1%, superior to the results of 91.3% obtained from the principal component analysis. In [7], a variant of SVMs is presented for automated prostate cancer localization, and the results showed that the proposed method can significantly boost the performance in contrast to the traditional SVMs. In another study, Liu Ying [28] introduced a SVMs based active learning algorithm and then applied it to gene expression profiles of prostate cancer samples for classification. Compared with passive learning, the proposed algorithm yielded more accurate results in classifying cancerous samples.

Although various prostate cancer detection studies have been performed using SVMs and its variants, very few studies have been done to deal with class imbalance problem mainly. Class imbalance is a significant challenge in the cancer data in which there is a much larger number of normal cases compared to the cancer cases in a cohort [25]. Most traditional machine learning methods like SVMs directly training on the imbalanced datasets tend to be overwhelmed by the majority class and thus lead to the deterioration of the performance on the minority class [14]. In such scenarios, the Area Under the ROC Curve (AUC) has been recognized as a more appropriate performance index than accuracy [22]. Since AUC maximization becomes the target, a classifier that is designed to optimize AUC directly should have a significant advantage to solve the class imbalance problem. To achieve this goal, some efforts have been made to adapt SVMs for straight-forward AUC optimization [5,9,23,39,42].

This study mainly concentrates on how to integrate the AUC optimization into the SVMs based learning framework. More concretely, we propose a new AUC optimization algorithm called AUC-LS-SVMs based on a well-known variant of SVMs - least square support vector machines (LS-SVMs) [38] to particularly handle the class imbalance problem in the early prostate cancer detection. LS-SVMs have the comparable classification performance to the classical SVMs, but their learning processes are greatly simplified by solving a set of linear equations instead of a QP problem in SVMs [38]. More importantly, with the proposed generalized AUC performance index $R_{AUCLS}$, the analytical solution of AUC-LS-SVMs can facilitate the fast leave-one-out cross validation (LOOCV) error estimate for AUC's parameter tuning, which greatly reduces the computational cost. To sum up, the contributions of our work on the proposed classifier AUC-LS-SVMs are as follows.

- A generalized AUC performance index $R_{AUCLS}$ is proposed by introducing the parameter $\lambda$. Our theoretical analysis indicated that $R_{AUCLS}$ is linearly dependent on the classical AUC performance index $R_{AUC}$.
- With both $R_{AUCLS}$ and LS-SVMs, AUC-LS-SVMs is proposed to have an analytical solution to explicitly handle the class imbalance problem. Differ from the traditional cost sensitive methods, AUC-LS-SVMs does not require prior knowledge of the misclassification costs in advance.
- A fast leave-one-out cross validation strategy for the parameter $\lambda$ in $R_{AUCLS}$ is developed to guarantee that of AUC-LS-SVMs is an almost unbiased classifier on imbalanced data, without heavy computational burden.

The rest of the paper is organized as follows. Section 2 introduces the related work about SVMs based methods for class imbalance problems. Section 3 clarifies the notations used in the paper, reviews LS-SVM and states the proposed concept of $R_{AUCS}$. Section 4 presents the proposed classifier AUC-LS-SVMs. Section 5 states the proposed fast leave-one-out cross validation strategy for the proposed classifier. Experimental results on the public datasets and a real-world prostate cancer dataset are presented and analyzed in Section 6. Section 7 concludes the paper.

## 2 Related Work

Here we briefly review the related works about SVMs based methods for class imbalance problems. Commonly, the performance of SVMs and its variants deteriorates due to imbalanced class distributions. A standard solution is to assign different error costs to training samples based on their classes [18,26,29,40,43]. For example, Zhu et al. [44] proposed a majority projection (MP) version of the traditional multiple empirical kernel learning (MEKL) to deal with imbalanced problems by introducing a weight matrix and a regularization term into MEKL. In another study, Ghazikhani et al. [21] developed an on-line model to deal with concept drift and class imbalance in which different importance to error is assigned in separate classes. These SVMs based approaches are based on the assumption that the misclassification costs are already known [18,26]. However, we usually do not have prior information regarding these costs; thus, we cannot correctly use these approaches. Another solution to deal with class imbalance issues is to incorporate AUC optimization in SVMs based methods. The ROC curve is an appropriate metric to evaluate machine learning models on imbalanced datasets, representing the rates of true positive against false positive under different thresholds. AUC refers to the area under ROC curve. It is equal to the probability that a classifier assigns a higher score to a randomly chosen positive instance than a randomly chosen negative one. Higher the AUC is, better the model is at distinguishing two imbalanced classes. Since the goal of imbalanced data classification is to find a decision function having high AUC value, it is natural to consider modifying a learning algorithm that can directly maximize AUC performance index. In literature, some AUC optimization studies using SVMs have been proposed [8, 16,9,24,31,42]. For example, Rakotomamonjy [31] proposed a SVMs based algorithm for AUC maximization in which a numerically tractable approximation of AUC criterion is derived. To speed up the QP problem, a subset $m$ of interesting neighbors of a sample can be user-defined. However, the best value of $m$ is problem-dependent due to the relevance to class distributions. Scheffer [8] proposed an AUC maximizing SVMs with $O(n^4)$ time complexity which becomes feasible only for small datasets. The proposed k-means AUC SVMs is more efficient in which the running time is quadratic in the sample size but only feasible for linear kernels.

Since the introduction of SVMs, various variants have been developed. For example, in [32], the authors combined the SVM algorithms with intuitionistic fuzzy sets for the first time to build a IFTSVM model which significantly alleviates the influence of noises and outliers. In this study, we focus on one of the classical variants of SVMs, i.e., least-squares SVMs (LS-SVMs). It simplifies the SVMs' formulation without losing the classification performance advantages. Moreover, LS-SVMs can formulate a fast leave-one-out cross validation strategy to unbiasedly estimate the actual generalization ability in parameter tuning procedures and reduce high computational cost [12]. Thus, in this study, we seek for a LS-SVMs based AUC method for imbalanced data classification to improve both effectiveness and efficiency. To our knowledge, this is the first work to develop a LS-SVMs method on the imbalanced data, which can directly maximize the AUC and use the fast leave-one-out cross validation strategy to guarantee an almost unbiased estimate in parameter tuning.

## 3 On LS-SVMs and the proposed performance index $R_{AUCLS}$

In this section, we will briefly review the well-known least square support vector machine (LS-SVM) and the concept of AUC, and accordingly, propose the generalized AUC performance index $R_{AUCLS}$.

### 3.1 On LS-SVMs for binary classification

The LS-SVMs classifier [34] is proposed as an alternative formulation compared to the traditional SVMs. Given the training set $S$ of $N$ samples $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the $i$-th sample and $y_i \in \{+1, -1\}$ is the corresponding label. The discriminant function has the following form:

$$g(\boldsymbol{x}) = \boldsymbol{w}^T \varphi(\boldsymbol{x}) + b \tag{1}$$

where $\varphi(\cdot) : \mathbb{R}^d \to \mathbb{R}^h$ is the mapping function that maps the sample $\boldsymbol{x}$ from the input space to the higher dimensional feature space. After $\boldsymbol{w}$, $b$ are fixed, the label of a testing sample $\boldsymbol{x}$ can be easily predicted according to the sign of $g(\boldsymbol{x})$.

The LS-SVMs classifier is obtained by finding the solution of the following primal optimization problem

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{\gamma}{2}\sum_{i=1}^{N}\xi_i^2 \\ \text{s.t} \quad & y_i = \boldsymbol{w}^T\varphi(\boldsymbol{x}_i) + b + \xi_i, \quad i = 1, 2, ..., N \end{aligned} \tag{2}$$

where the trade-off parameter $\gamma > 0$. In particular, the dual optimization problem of Eq. (2) has an analytical solution [34], which can easily be used to formulate the fast LOOCV error estimate for parameter tuning so as to reduce the computational cost.

### 3.2 On AUC, $R_{AUC}$ and $R_{AUCLS}$

Given an input space $X$, an output space $Y = \{-1, 1\}$, and P as the distribution of $(\boldsymbol{x}, y)$, i.e. the joint distribution of an sample $\boldsymbol{x}$ and its corresponding label $y$. The target of a classification algorithm is to learn a scoring function $s$ as a map $X \to \mathbb{R}$, where $s(\boldsymbol{x}_i)$ is the scoring function which is in proportion to $P(y = 1|\boldsymbol{x})$, which is the conditional probability of the label to be positive given its inputs.

As stated in [9,23,42,39,16,19,20], AUC refers to the possibility that a randomly sampled positive sample has a higher score than a negative one, which can be represented using

$$\mathrm{AUC} = \mathrm{P}(\mathrm{s}(\boldsymbol{x}_1) > \mathrm{s}(\boldsymbol{x}_2)|\mathrm{y}_1 = 1, \mathrm{y}_2 = -1) \tag{3}$$

It has been known that AUC is a more stable and robust performance index than accuracy for imbalanced binary classification tasks. However, since we usually do not know the distribution P, AUC cannot be directly calculated from Eq. (3). In order to address this issue, given a training dataset $S$ of $N$ samples sampled from P, i.e., $S = \{(\boldsymbol{x}_i, y_i) \in (X \times Y),\ i = 1, \cdots, N\}$, we can estimate AUC by substituting the possibility with its corresponding frequency on $S$. The loss form of this estimation can be expressed as

$$\overline{\mathrm{AUC}} = \sum_{i \in N_+} \sum_{j \in N_-} \frac{I[s(\boldsymbol{x}_i) < s(\boldsymbol{x}_j)] + \frac{1}{2} I[s(\boldsymbol{x}_i) = s(\boldsymbol{x}_j)]}{n_+ n_-} \tag{4}$$

where $I[A]$ is the indicator function. That is to say, $I[A] = 1$ if and only if $A$ is true, $I[A] = 0$ otherwise; $N_+$ is the set of all the indices of positive samples and $N_-$ is the set of those of the negative samples; $n_+ = |N_+|$; $n_- = |N_-|$.

Although we could estimate AUC on any dataset sampled from P, we notice that, the ranking loss

$$I[s(\boldsymbol{x}_i) < s(\boldsymbol{x}_j)] + \frac{1}{2} I[s(\boldsymbol{x}_i) = s(\boldsymbol{x}_j)], \quad \boldsymbol{x}_i \in N_+, \boldsymbol{x}_j \in N_- \tag{5}$$

is a discrete and non-differentiable function, straightforward optimization of $\overline{\mathrm{AUC}}$ is a *NP* problem. To make it practical, Eq. (5) is often approximated by a differentiable surrogate loss function $l(t)$. Replacing $l(t)$ into Eq. (4), we can obtain the classical surrogate empirical risk function $R_{AUC}$ [16,19,20] based on the training dataset $S$

$$R_{AUC} = \sum_{i \in N_+} \sum_{j \in N_-} \frac{l(s(\boldsymbol{x}_i) - s(\boldsymbol{x}_j))}{n_+ n_-} \tag{6}$$

As stated in [19], the surrogate loss $l(t)$ in Eq. (6) may be taken as $l(t) = (1-t)^2$ with the consistence guarantee of AUC. In other words, we have

$$R_{AUC} = \sum_{i \in N_+} \sum_{j \in N_-} \frac{1 - (s(\boldsymbol{x}_i) - s(\boldsymbol{x}_j))^2}{n_+ n_-} \tag{7}$$

Obviously, if we take the discriminant function in Eq. (1) as $s(\boldsymbol{x})$ in Eq. (7), then $b$ will automatically vanish in terms of $\big(s(\boldsymbol{x}_i) - s(\boldsymbol{x}_j)\big)$ in Eq. (7). As a result, in this study, we take $s(\boldsymbol{x})$ in Eq. (7) as the discriminant function in Eq. (1), and then propose the generalized AUC performance index $R_{AUCLS}$ by introducing the parameter $\lambda$ in $R_{AUC}$. That is to say, $R_{AUCLS}$ is defined as

$$\begin{aligned} R_{AUCLS} &= \sum_{i \in N_+} \sum_{j \in N_-} \frac{(\lambda - (\boldsymbol{w}^T \varphi(\boldsymbol{x}_i) + b - \boldsymbol{w}^T \varphi(\boldsymbol{x}_j) - b))^2}{n_+ n_-} \\ &= \sum_{i \in N_+} \sum_{j \in N_-} \frac{(\lambda - (\boldsymbol{w}^T (\varphi(\boldsymbol{x}_i) - \varphi(\boldsymbol{x}_j)))^2}{n_+ n_-} \end{aligned} \tag{8}$$

To best of our knowledge, up to date, no effort has been taken to generalize the concept of AUC in this way.

**Theorem 1** *For a binary classification task, when $s(\boldsymbol{x})$ in Eq. (7) is taken as the discriminant function in Eq. (1), minimizing $R_{AUCLS}$ is equivalent to minimizing $R_{AUC}$ with another discriminant function which is just the discriminant function in $R_{AUCLS}$ multiplied by a constant.*

*Proof* When $s(\boldsymbol{x})$ takes the discriminant function in Eq. (1), $R_{AUC}$ in Eq. (7) becomes Eq. (8). After multiplying the discriminant function with a constant $\lambda$, we can readily know that the new discriminant function after such a multiplication does not change the discriminant result for a binary classification task. Therefore, with such a $\lambda$, the discriminant function becomes $\lambda(\boldsymbol{w}^T \varphi(\boldsymbol{x}_i) + b)$. As $R_{AUCLS}$ is not dependent on $b$, $R_{AUCLS}$ with the new discriminant function becomes

$$\begin{aligned} R_{AUCLS} &= \sum_{i \in N_+} \sum_{j \in N_-} \frac{(\lambda - \lambda \boldsymbol{w}^T (\varphi(\boldsymbol{x}_i) - \varphi(\boldsymbol{x}_j)))^2}{n_+ n_-} \\ &= \lambda^2 \sum_{i \in N_+} \sum_{j \in N_-} \frac{(1 - \boldsymbol{w}^T (\varphi(\boldsymbol{x}_i) - \varphi(\boldsymbol{x}_j)))^2}{n_+ n_-} \\ &= \lambda^2 R_{AUC} \end{aligned} \tag{9}$$

When $\lambda$ is given, $\lambda^2$ becomes a constant. Thus, this theorem holds true.

Theorem 1 reveals that without any change in the discriminant results on a binary classification task, when

$R_{AUCLS}$ achieves the minimum, $R_{AUC}$ achieves its minimum as well. Here, the introduction of the parameter $\lambda$ contributes to the development of the fast leave-one-out cross validation strategy for the proposed classifier, which will be discussed in the next section.

## 4 The proposed classifier

In this section, we will introduce the proposed classifier AUC-LS-SVMs for binary class imbalance learning. To achieve this, according to Theorem 1, we directly add the proposed generalized AUC performance index $R_{AUCLS}$ in Eq. (8) to the LS-SVMs framework. Thus, we have the following primal optimization problem

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{C}{2}R_{AUCLS} + \frac{\gamma}{2}\sum_{i=1}^{N}\xi_i^2 \\ \text{s.t} \quad & y_i = \boldsymbol{w}^T\varphi(\boldsymbol{x}_i) + b + \xi_i, \quad i = 1, 2, ..., N \end{aligned} \tag{10}$$

Compared to the traditional LS-SVMs in Eq. (2), the proposed classifier in Eq. (10) has an additive term $R_{AUCLS}$. Since the additive term $R_{AUCLS}$ is essentially a convex function about $\boldsymbol{w}$, $R_{AUCLS}$ actually contributes an extra generalization capability to the traditional LS-SVMs, according to the statistical learning theory [35]. In other words, the proposed classifier in Eq. (10) has more generalization capability than the traditional LS-SVMs. This will also be proved experimentally on the imbalanced datasets in Section 6 later. In addition, the distinctive benefit from the use of $R_{AUCLS}$ in Eq. (9) exists in that it can help us derive a fast leave-one-out cross validation strategy for the proposed classifier AUC-LS-SVMs, which will be clearly seen in the next section.

According to the Lagrangian optimization strategy, Eq. (10) has its dual solution. That is to say, by using the matrix inversion, the model parameter $\boldsymbol{\alpha}$ and $b$ in the corresponding dual optmization problem can be simply calculated by the following Eq. (11).

$$\begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \mathbf{P}\begin{bmatrix} \boldsymbol{y} - \lambda\boldsymbol{f} \\ 0 \end{bmatrix} \tag{11}$$

where $\mathbf{P} = \mathbf{V}^{-1}$ and $\mathbf{V}$ is the first matrix on the left in Eq. (36) in the appendix. Therefore, the detailed derivations can be found in the appendix. The final discriminant function of the proposed AUC-LS-SVMs can be expressed as follows

$$\begin{aligned} g(\boldsymbol{x}) &= \varphi^T(\boldsymbol{x})\boldsymbol{w} + b \\ &= \Big(\varphi^T(\boldsymbol{x}) - \frac{\varphi^T(\boldsymbol{x})}{M}\sum_{k\in N^+}\sum_{l\in N^-}\big(\varphi(\boldsymbol{x}_k) - \varphi(\boldsymbol{x}_l)\big)\big(\varphi(\boldsymbol{x}_k) - \varphi(\boldsymbol{x}_l)\big)^T\Big) \\ &\quad \Big(\sum_{i=1}^{N}\alpha_i\varphi(\boldsymbol{x}_i) + \frac{\lambda C}{n^+n^-}\sum_{i\in N^+}\sum_{j\in N^-}\big(\varphi(\boldsymbol{x}_i) - \varphi(\boldsymbol{x}_j)\big)\Big) + b \\ &= \Big(\varphi^T(\boldsymbol{x}) - \frac{1}{M}\sum_{k\in N^+}\sum_{l\in N^-}\big(k(\boldsymbol{x},\boldsymbol{x}_k) - k(\boldsymbol{x},\boldsymbol{x}_l)\big)\big(\varphi(\boldsymbol{x}_k) - \varphi(\boldsymbol{x}_l)\big)^T\Big) \\ &\quad \Big(\sum_{i=1}^{N}\alpha_i\varphi(\boldsymbol{x}_i) + \frac{\lambda C}{n^+n^-}\sum_{i\in N^+}\sum_{j\in N^-}\big(\varphi(\boldsymbol{x}_i) - \varphi(\boldsymbol{x}_j)\big)\Big) + b \\ &= \sum_{i=1}^{N}\alpha_i k(\boldsymbol{x},\boldsymbol{x}_i) + \frac{\lambda C}{n^+n^-}\sum_{i\in N^+}\sum_{j\in N^-}\big(k(\boldsymbol{x},\boldsymbol{x}_i) - k(\boldsymbol{x},\boldsymbol{x}_j)\big) \\ &\quad - \frac{1}{M}\sum_{k\in N^+}\sum_{l\in N^-}\big(k(\boldsymbol{x},\boldsymbol{x}_k) - k(\boldsymbol{x},\boldsymbol{x}_l)\big)\sum_{i=1}^{N}\alpha_i\big(k(\boldsymbol{x}_k,\boldsymbol{x}_i) - \\ &\quad k(\boldsymbol{x}_l,\boldsymbol{x}_i)\big) - \frac{\lambda C}{Mn^+n^-}\sum_{k\in N^+}\sum_{l\in N^-}\sum_{i\in N^+}\sum_{j\in N^-}\big(k(\boldsymbol{x},\boldsymbol{x}_k) - k(\boldsymbol{x},\boldsymbol{x}_l)\big) \\ &\quad \big(k(\boldsymbol{x}_k,\boldsymbol{x}_i) - k(\boldsymbol{x}_k,\boldsymbol{x}_j) - k(\boldsymbol{x}_l,\boldsymbol{x}_i) + k(\boldsymbol{x}_l,\boldsymbol{x}_j)\big) + b \end{aligned} \tag{12}$$

With Eq. (11) and Eq. (12), we summarize the entire learning algorithm of AUC-LS-SVMs in Algorithm 1. Please note, Algorithm 1 uses a fast leave-one-out cross validation strategy to tune the parameter $\lambda$, which will be stated in algorithm 2 in the next section.

**Algorithm 1** Learning Algorithm of AUC-LS-SVMs

1: **Input** training set of $N$ samples $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$, $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \{+1, -1\}$ for binary classification, and the trade-off parameters $\gamma \in \{1, 10, 50, 100, 150, 200, 250\}$, $C \in \{0.01, 0.1, 1, 10, 25, 50\}$ in our experiments
2: **Calculate** the kernel matrix $\tilde{\mathbf{K}}$ and the vector $\boldsymbol{f}$ according to their definitions in Eq. (36) in the appendix
3: **Invoke** Algorithm 2 to obtain $\lambda$, where Algorithm 2 using the proposed fast LOOCV is given in the section 5
4: **Calculate** $\boldsymbol{w}$ and $b$ using Eq. (11) and Eq. (26) in the appendix
5: **Output** discriminant function $g(\boldsymbol{x})$ using Eq. (12)

When $\lambda$ is fixed, below let us discuss the computational complexity of both training and prediction of the above LS-SVMs. As for its training, according to Eq. (11) and Eq. (36) in the appendix, we first compute $\tilde{\mathbf{K}}$ in $\mathbf{V}$, which requires $O((n^+n^- + 1)N^2)$ in terms of the definition of $\tilde{k}(\boldsymbol{x}_i, \boldsymbol{x}_k)$ in the appendix. We then compute $\boldsymbol{f}$, which requires $O(N(n^+n^-)^3)$ in terms of the definition of $f(\boldsymbol{x}_i)$ in the appendix. Since computing $\mathbf{P} = \mathbf{V}^{-1}$ in Eq. (11) indeed requires $O(N^3)$, the computational complexity of training the above AUC-LS-SVMs becomes $O(N^3 + (n^+n^- + 1)N^2 + (n^+n^-)^3N)$. However, since $\tilde{\mathbf{K}}$, $\boldsymbol{f}$ can be computed only once in advance, and accordingly $\mathbf{P} = \mathbf{V}^{-1}$ can be computed in

advance, such a heavy computational burden will give no trouble for the use of the proposed LOOCV strategy in the next section.

To predict an unseen testing sample $\boldsymbol{x}$, it requires $O(N+n^+n^-+n^+n^-(N+(n^+n^-)^2))=O(N+(N+1)n^+n^-+(n^+n^-)^3)$ according to Eq. (12), which is time-consuming especially when $N$ is big and $n^+ \gg n^-$. Since Eq. (12) is a mixture of kernel functions, we may refer to the work in [41] to simplify our model's mixture expression to achieve a faster prediction on request in the future.

## 5 Fast leave-one-out cross validation strategy for parameter tuning

From Section 4, we can observe that the classification performance of the proposed classifier AUC-LS-SVMs surely relies on the values of parameters $\lambda$, $\gamma$ and $C$. To determine the optimal parameter $\lambda$ with the given parameters $\gamma$ and $C$, the classical leave-one-out cross validation (LOOCV) strategy as an almost unbiased estimator of the generalization error is taken here to minimize the leave-one-out error.

LOOCV is the extreme case of $K$-fold CV with $K$ equal to the number $N$ of samples in the dataset. In each iteration during LOOCV, we train a model on all the data except for one remained for prediction. This process is repeated $N$ times to ensure that every sample from the training dataset has the same chance for model construction and evaluation. The average leave-one-out error is computed from those $N$ constructed models for performance. However, LOOCV is a computationally expensive procedure. To overcome this shortage, referring to our recent works [37,36], here we derive a fast LOOCV strategy for the proposed classifier AUC-LS-SVMs to tune the parameter $\lambda$ in Eq. (11).

We decompose $\mathbf{V}$ into its block presentation with the isolation of the first row and column as follows:

$$\mathbf{V}=\begin{bmatrix}\tilde{\mathbf{K}}+\frac{\boldsymbol{I}}{\gamma} & \mathbf{1}\\ \mathbf{1}^T & 0\end{bmatrix}=\begin{bmatrix}v_{11} & \mathbf{v}_1^T\\ \mathbf{v}_1 & \mathbf{V}_{(-1)}\end{bmatrix} \tag{13}$$

We denote $\boldsymbol{\alpha}_{(-i)}$ and $b_{(-i)}$ as the model parameters during the $i$-th iteration of the leave-one-out cross validation. In the first iteration, we have:

$$\begin{bmatrix}\boldsymbol{\alpha}_{(-1)}\\ b_{(-1)}\end{bmatrix}=\mathbf{P}_{(-1)}\left(\boldsymbol{y}_{(-1)}-\lambda f(\boldsymbol{x}_{(-1)})\right) \tag{14}$$

where $\mathbf{P}_{(-1)}=\mathbf{V}_{(-1)}^{-1}$ and $y_{(-1)}=[y_2;y_3;\cdots;y_N;0]$. We denote the predicted label of the $i$-th sample excluded from the training dataset as $\tilde{y}_i$. The predicted label of the first training sample can be represented as

$$\begin{aligned}\tilde{y}_1&=\mathbf{v}_1^T\begin{bmatrix}\boldsymbol{\alpha}_{(-1)}\\ b_{(-1)}\end{bmatrix}+\lambda f(\boldsymbol{x}_{(-1)})\\&=\mathbf{v}_1^T\mathbf{P}_{(-1)}\left(\boldsymbol{y}_{(-1)}-\lambda f(\boldsymbol{x}_{(-1)})\right)+\lambda f(\boldsymbol{x}_{(-1)})\end{aligned} \tag{15}$$

Considering the last $N$ equations in Eq. (36), we obtain $\left[\mathbf{v}_1\ \mathrm{V}_{(-1)}\right]\left[\boldsymbol{\alpha}^T,\ b\right]^T=(\boldsymbol{y}_{(-1)}-\lambda f(\boldsymbol{x}_{(-1)})$, and therefore Eq. (15) can be further written into

$$\begin{aligned}\tilde{y}_1&=\mathbf{v}_1^T\mathbf{P}_{(-1)}\left[\mathbf{v}_1\ \mathbf{V}_{(-1)}\right][\alpha_1,\cdots,\alpha_N,b]^T+\lambda f(\boldsymbol{x}_{(-1)})\\&=\mathbf{v}_1^T\mathbf{P}_{(-1)}\mathbf{v}_1\alpha_1+\mathbf{v}_1^T[\alpha_2,\cdots,\alpha_N,b]^T+\lambda f(\boldsymbol{x}_{(-1)})\end{aligned} \tag{16}$$

In Eq. (36), the first equation of the system is $y_1-\lambda f(\boldsymbol{x}_{(-1)})=v_{11}\alpha_1+\mathbf{v}_1^T[\alpha_2,\alpha_3,\cdots,\alpha_N,b]^T$. Combined with Eq. (16), we obtain $\tilde{y}_1=y_1-\lambda f(\boldsymbol{x}_{(-1)})$. Lastly, by using $\mathbf{P}=\mathbf{V}^{-1}$ and the block matrix inversion lemma, we can obtain

$$\mathbf{P}=\begin{bmatrix}u^{-1} & -u^{-1}\mathbf{v}_1\mathbf{P}_{-1}\\ \mathbf{P}_{(-1)}+u^{-1}\mathbf{P}_{(-1)}\mathbf{v}_1^T\mathbf{v}_1\mathbf{P}_{(-1)} & -u^{-1}\mathbf{P}_{(-1)}\mathbf{v}_1^T\end{bmatrix} \tag{17}$$

where $u=v_{11}-\mathbf{v}_1^T\mathbf{P}_{(-1)}\mathbf{v}_1$. Since the system of linear equations in Eq. (36) is not sensitive to the permutations of the ordering of the equations, we obtain

$$\tilde{y}_i=y_i-\alpha_i/P_{ii} \tag{18}$$

By defining $\left[\boldsymbol{\alpha}^{'T},b^{'}\right]^T=\mathbf{P}\left[\boldsymbol{y}^T,0\right]$, $\left[\boldsymbol{\alpha}^{''T},b^{''}\right]=\mathbf{P}\left[\boldsymbol{f}^T,0\right]$, and $\boldsymbol{\alpha}=\boldsymbol{\alpha}^{'}-\lambda\boldsymbol{\alpha}^{''}$, then we obtain

$$\tilde{y}_i=y_i-\frac{\alpha_i^{'}}{P_{ii}}+\frac{\lambda\alpha_i^{''}}{P_{ii}} \tag{19}$$

It is assumed that the optimal $\lambda$ will retain the same signs of $\tilde{y}_i$ and $y_i$ for all the samples in the training dataset. However, this might result in local minima issues due to its non-convex formulation. Thus, we use the following loss function similar to hinge loss function:

$$l(\tilde{y}_i,y_i)=|1-\tilde{y}_iy_i|_+=\left|y_i\frac{\alpha_i^{'}-\lambda\alpha_i^{''}}{P_{ii}}\right|_+ \tag{20}$$

where $|x|_+=\max\{0,x\}$. This is a convex upper bound to the leave-one-out misclassification loss. It prefers the solutions in which $\tilde{y}_i$ retains the same sign of $y_i$ with an absolute value equal to or bigger than 1. Moreover,

considering that the cost of minority class may be overlooked by the majority class, we enlarge minority class' cost by $\frac{n_+}{n_-}$. Therefore, the objective function becomes:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n^+} l(\tilde{y}_i, y_i) + \frac{n^+}{n^-} \sum_{i=1}^{n^-} l(\tilde{y}_i, y_i) \\ \text{s.t} \quad & 0 \leq \lambda \leq D \end{aligned} \tag{21}$$

where $D$ is a constant. This optimization process can be implemented by a projected sub-gradient descent algorithm, whose pseudo-code is given in Algorithm 2.

**Algorithm 2** Projected Sub-gradient Descent Algorithm

1: **Input** $\boldsymbol{\alpha}^{'}$, $\boldsymbol{\alpha}^{''}$;
2: **Initialize** $\lambda(0) \leftarrow 0$, $t \leftarrow 1$ and $\varepsilon = 10^{-3}$;
**repeat**
  **for** $i = 1, 2, \ldots, N$ **do**
    3: **Update** $\tilde{y}_i = y_i - \frac{\alpha_i^{'}}{\boldsymbol{P}_{ii}} + \frac{\lambda \alpha_i^{''}}{\boldsymbol{P}_{ii}}$;
    4: **If** $\tilde{y}_i y_i > 0$:
      $d_i = 1$
    **Else** $d_i = 0$
  **end**
  5: **Update** $\lambda(t) \leftarrow \lambda(t-1) - \frac{1}{\sqrt{t}} \Big[ \sum_{i=1}^{n^+} d_i y_i \frac{\alpha_i^{''}}{P_{ii}} + \sum_{i=1}^{n^-} d_i y_i \frac{n^+}{n^-} \frac{\alpha_i^{''}}{P_{ii}} \Big]$;
  6: **If** $\lambda(t) > D$:
    $\lambda(t) \leftarrow D$
  **Else** $\lambda(t) \leftarrow \max(\lambda(t), 0)$
  7: **Update** t = t + 1;
**until** $|\lambda(t+1) - \lambda(t)| \leq \varepsilon$;
8: **Output** $\lambda(t)$ **as** $\lambda$;

One highlight of the proposed classifier AUC-LS-SVMs is its fast computational ability in finding the optimal value of $\lambda$ using the proposed LOOCV strategy. Below let us analyze the reason for this highlight. Eq. (19) is our fast LOOCV evaluation formula. It is easy to see from Eq. (19) that once we get $\mathbf{P}$ and $\boldsymbol{y}$ for the whole training dataset, the solution for the $i$ iteration in the LOOCV can be calculated. This is much faster as compared to the direct calculation from $\mathbf{P}_{i(-1)}$ to get $\tilde{y}_i$. Therefore, our proposed LOOCV only needs to calculate the matrix inversion once. In contrast, the traditional LOOCV requires to calculate the matrix inversions for $N$ times.

In detail, our fast LOOCV's computational cost can be represented as $O(N^3 + N)$, where $O(N^3)$ is due to the calculation of matrix $\mathbf{P}$ by the inverse of $\mathbf{V}$ related to the training set, and $O(N)$ is due to the $N$ iterations in Algorithm 2 for optimizing Eq. (21).

However, if we use the traditional LOOCV with grid search, we need to find the optimal $\lambda$ from a range $[\lambda_1, \lambda_2, ..., \lambda_T]$. The whole computational complexity would become $T * O(N^3 * N) = T * O(N^4)$. This is much more computationally expensive than the proposed fast LOOCV strategy here.

Table 1: Details of UCI public datasets

| Dataset | # samples | # dimensions | IR |
|---|---|---|---|
| *pima* | 768 | 8 | 1.8657 |
| *ILPD* | 579 | 10 | 0.3986 |
| *liver* | 345 | 6 | 1.3793 |
| *haerman* | 306 | 3 | 2.7778 |
| *german* | 1000 | 24 | 2.3333 |
| *australian* | 690 | 14 | 1.2476 |
| *svmguide1* | 3089 | 4 | 0.5445 |
| *svmguide3* | 1243 | 22 | 3.1993 |

## 6 Experimental results

The focus of this study is to handle imbalanced prostate cancer datasets effectively by combining the AUC optimization into an SVM based framework (i.e., LS-SVMs) with fast leave-one-out cross validation. Therefore, we take both SVMs and LS-SVMs as the comparative methods in our experiments. Although OPAUC in [19] seems to be related to AUC-LS-SVMs, it is for on-line learning, which is another subfield of the research domain and therefore is not taken as a comparative method. The performances of AUC-LS-SVMs are evaluated on various public datasets and a real-world prostate cancer dataset.

### 6.1 Public datasets

In this section, in order to examine the performance of the proposed classifier AUC-LS-SVMs in comparison with traditional methods, we conduct extensive experiments on various benchmark datasets. Table 1 shows the details of eight binary-class datasets used in our experiments. *pima, ILPD, liver, haerman, german and australian* datasets can be downloaded from UCI Machine learning Repository [4]. *svmguide1* and *svmguide3* can be downloaded from LIB-SVM [3]. # samples and # dimensions represent the number of samples and the number of dimensions, respectively. IR refers to the imbalance ratio:

$$\text{IR} = \frac{n^-}{n^+} \tag{22}$$

where $n^-$ is the size of the negative class set and $n^+$ is that of the positive class.

To make our comparison fair, we adopt the same setup for all methods. Gaussian kernel $k(\boldsymbol{x}, \boldsymbol{y}) = exp(-\frac{||\boldsymbol{x}-\boldsymbol{y}||^2}{2\sigma^2})$ is used for AUC-LS-SVMs and the comparative methods. For AUC-LS-SVMs, please note that our proposed

fast LOOCV is used to tune the trade-off parameter $\lambda$ given the parameters $\gamma$ and $C$. Therefore, three parameters still need to be defined in advance, which are $\gamma$, $C$ and kernel parameter $\sigma$. Grid search with cross validation is used to determine their optimal values from the sets $\{1, 10, 50, 100, 150, 200, 250\}$, $\{0.01, 0.1, 1, 10, 25, 50\}$ and $\{2e-5, 2e-4, 2e-3, 2e-2, 2e-1, 1, 2e1, 2e2\}$, respectively. For SVMs and LS-SVMs, grid search with cross validation is used to tune the kernel parameter $\sigma$ and trade-off parameter $C$ from $\{2e-5, 2e-4, 2e-3, 2e-2, 2e-1, 1, 2e1, 2e2\}$ and $\{1, 10, 50, 100, 150, 200, 250\}$, respectively.

10-fold cross-validation is used to evaluate the classification performance of all the methods by partitioning the original dataset into nine folds of training subset and one testing subset for evaluation. AUC and F1-score (i.e., the harmonic mean of precision and recall) are selected as the evaluation metrics due to their sensitivity to class imbalance. The other commonly used metrics such as accuracy, precision, recall are measured as well for more performance observations in our experiments. All the experiments are implemented using MATLAB R2014a on a computer with Intel Core i7-4700MQ 2.40 GHz CPU and 8.00GB RAM.

Table 6 shows the average classification performance of AUC-LS-SVMs and the comparative methods after 10-fold cross validation on the public datasets. To see if there are significant differences among the performances of the proposed classifier and comparative methods over these datasets, we employ the Friedman ranking test. From Tables 2 and 4, the null hypothesis that all the methods perform the same in terms of accuracy (p=0.009804) and AUC (p=0.002187) on average is rejected. Thus, a Holm post-hoc test is carried out to compare AUC-LS-SVMs with the other two methods further. We set $\alpha = 0.05$ as the level of confidence in all cases. According to results from Tables 3 and 5, AUC-LS-SVMs significantly outperforms the traditional methods LS-SVMs and SVMs in terms of both accuracy and AUC in our experiments. Hence, these results allow concluding that when model selection is performed appropriately, our proposed classifier AUC-LS-SVMs can indeed maximize AUC. Besides, the accuracy results indicate that the proposed classifier AUC-LS-SVMs can achieve better AUC without sacrificing classification accuracy. Moreover, the experimental results show that the term $R_{AUCLS}$ in the objective function of the proposed classifier does help improve the generalization capability on the testing datasets compared to the traditional LS-SVMs. In terms of precision and recall, AUC-LS-SVMs achieves the highest precision values on all the datasets except *pima* and *svmguide3,* and the highest recall values on all the datasets except *ILPD* and *german.* We then take a look at F1 score, which incorporates precision and recalls equally. The experimental results show that AUC-LS-SVMs still achieves higher or at least comparable F1-score over different datasets, showing a good prediction power on imbalanced datasets.

In general, we can see that AUC-LS-SVMs achieves excellent classification performances on the public datasets with different imbalanced class distributions. We believe that our proposed classifier distinguishes from the comparative methods in two aspects. One is the AUC-based objective function. The advantage of AUC-LS-SVMs is mainly due to the direct use of the proposed generalized AUC performance index $R_{AUCLS}$ as a part of the objective function for training. Thus, we can construct a reliable prediction model directly on the imbalanced dataset instead of going through an additional data preprocessing step. The second aspect is the fast leave-one-out cross validation for parameter tuning. By using the proposed strategy, the parameter $\lambda$ can be tuned quickly and autonomously.

Table 2: Average rankings of AUC-LS-SVMs and the comparative methods in terms of accuracy ($p$-value=0.009804)

| Methods | Ranking |
|---|---|
| AUC-LS-SVMs | 1.125 |
| LS-SVMs | 2.375 |
| SVMs | 2.5 |

Table 3: Holm post-hoc comparison results for AUC-LS-SVMs and the other methods in terms of accuracy with $\alpha = 0.05$

| $i$ | Methods | $z$-value | $p$-value | Holm=$\alpha/i$ |
|---|---|---|---|---|
| 2 | SVMs | 2.75 | 0.00596 | 0.025 |
| 1 | LS-SVMs | 2.5 | 0.012419 | 0.05 |

Table 4: Average rankings of AUC-LS-SVMs and the comparative methods in terms of AUC ($p$-value=0.002187)

| Methods | Ranking |
|---|---|
| AUC-LS-SVMs | 1 |
| LS-SVMs | 2.625 |
| SVMs | 2.375 |

Table 5: Holm post-hoc comparison results for AUC-LS-SVMs and the other methods in terms of AUC with $\alpha = 0.05$

| $i$ | Methods | $z$-value | $p$-value | Holm=$\alpha/i$ |
|---|---|---|---|---|
| 2 | LS-SVMs | 3.25 | 0.001154 | 0.025 |
| 1 | SVMs | 2.75 | 0.00596 | 0.05 |

6.2 Real-world prostate cancer dataset

A real-world prostate cancer dataset is used in this study, which is retrieved from a TRUS-guided prostate biopsy database in a hospital in Hong Kong. In total, there are 5899 patient records after TRUS-guided prostate biopsy. The patient information include 'any previous biopsy', 'age', 'PSA', 'DRE finding', 'DRE volume', 'TRUS volume', 'abnormal TRUS findings', 'pathology of TRUS', and 'total gleason score'. Referring to the ERSPC risk calculator [2] and consultations with local urologists, we plan to build two prostate cancer diagnostic models using different groups of features. The first model (i.e., *model (1)* in brevity) is built using the features - age, PSA, DRE finding, TRUS volume and abnormal TRUS finding, while the second model (i.e., *model (2)* in brevity) is built using the features - age, PSA, DRE finding and DRE volume.

Both models target patients who just had their initial biopsy. The patients who previously had biopsies before are excluded from this study. Case deletion is adopted to remove the patient records with any missing value(s). After that, feature scaling is applied to normalize the range of features into $[0, 1]$. The baseline characteristics of the processed datasets for model (1) and model (2) are presented in Tables 9 and 10, respectively. The distribution of two classes under 'outcome of biopsy' is very imbalanced, where 'non-cancer or insignificant cancer' versus 'significant cancer' is 5.9254:1 and 7.8874:1 in two cases, respectively. The normal cases dominate over the cancer cases, although in the clinical practice, it weighs more heavily on the detection of cancer than non-cancer.

To solve such class imbalance problem, we apply the proposed classifier AUC-LS-SVMs to build model (1) and model (2) on the processed datasets, and further evaluate and compare the performance with the traditional SVMs [13] and LS-SVMs [34]. The same parameter setting in Section 6.2 is used here for the proposed and comparative methods. Tables 7 and 8 show the average classification performance of AUC-LS-SVMs and the comparative methods after 10-fold cross validation for model (1) and model (2), respectively.

The experimental results show that SVMs achieved the highest accuracy (0.9377) and precision (0.8083) among all the methods for model (2). However, the proposed classifier AUC-LS-SVMs is superior to LS-SVMs and SVMs for AUC and F1-score in both model (1) and model (2). Here, it is important to remember that F1 score and AUC metrics are doing better than accuracy for skewed datasets. Therefore, we can conclude that in general, the proposed classifier AUC-LS-SVMs with direct AUC optimization have an advantage over the other methods in imbalanced prostate cancer detection.

## 7 Conclusions

Using pre-biopsy data to detect early prostate cancer can avoid unnecessary biopsies and overtreatment of low-grade prostate cancer. To construct a reliable prediction model for early prostate cancer detection, SVMs and their variants have been attracting more and more attention in this field. This study proposes a new classifier called AUC-LS-SVMs to explicitly deal with the class imbalance problem in prostate cancer detection. AUC-LS-SVMs directly integrates the proposed generalized AUC index $R_{AUCLS}$ into the objective function of LS-SVMs and uses the proposed fast LOOCV strategy to search for the best parameter $\lambda$ in $R_{AUCLS}$. Empirical results demonstrate that the proposed classifier outperformed the traditional methods LS-SVMs and SVMs for early prostate cancer detection. Extensive experiments are conducted on public datasets to confirm the efficacy of the proposed classifier further.

As stated in Section 4, AUC-LS-SVMs is comparatively time-consuming for prediction. In the future, we plan to develop simplified and even on-line versions to reduce the computational burden further. The proposed classifier can also be extended to deal with imbalanced multi-class classification, such as diagnosis of benign, insignificant, and significant prostate cancer.

Table 6: Evaluation on public datasets

| Datasets | Methods | | accuracy | precision | recall | F score | AUC |
|---|---|---|---|---|---|---|---|
| *pima* | AUC-LS-SVMs | training | 0.8043±0.0086 | 0.7753±0.0063 | 0.6484±0.0224 | 0.7060±0.0145 | 0.8994±0.0094 |
| | | testing | **0.7884±0.0226** | 0.6807±0.0964 | **0.6279±0.0460** | **0.6510±0.0555** | **0.8483±0.0291** |
| | LS-SVMs | training | 0.8407±0.0629 | 0.8286±0.0789 | 0.6911±0.1117 | 0.7524±0.0968 | 0.9063±0.0554 |
| | | testing | 0.7374±0.0439 | 0.6392±0.0735 | 0.5240±0.0630 | 0.5735±0.0567 | 0.8001±0.0532 |
| | SVMs | training | 0.7873±0.0122 | 0.7230±0.0250 | 0.6274±0.0170 | 0.6717±0.0183 | 0.8569±0.0076 |
| | | testing | 0.7576±0.0245 | **0.6814±0.0614** | 0.6056±0.1088 | 0.6335±0.0573 | 0.8169±0.0238 |
| *ILPD* | AUC-LS-SVMs | training | 0.7475±0.0114 | 0.7548±0.0140 | 0.9623±0.0085 | 0.8459±0.0080 | 0.8063±0.0105 |
| | | testing | **0.7137±0.0453** | **0.7265±0.0430** | 0.9507±0.0372 | 0.8228±0.0312 | **0.7595±0.0283** |
| | LS-SVMs | training | 0.8775±0.1259 | 0.8781±0.1245 | 0.9863±0.0194 | 0.9253±0.0757 | 0.9061±0.1050 |
| | | testing | 0.7063±0.0280 | 0.7222±0.0172 | 0.9562±0.0511 | 0.8222±0.0226 | 0.6454±0.0686 |
| | SVMs | training | 0.7194±0.0064 | 0.7239±0.0191 | 0.9868±0.0416 | 0.8343±0.0060 | 0.7551±0.0190 |
| | | testing | 0.7022±0.0201 | 0.7124±0.0193 | **0.9786±0.0675** | **0.8229±0.0197** | 0.6721±0.0302 |
| *liver* | AUC-LS-SVMs | training | 0.7668±0.0210 | 0.7752±0.0245 | 0.6312±0.0480 | 0.6953±0.0368 | 0.8056±0.0239 |
| | | testing | **0.7192±0.0478** | **0.6743±0.0676** | **0.6216±0.0878** | **0.6449±0.0663** | **0.7467±0.0526** |
| | LS-SVMs | training | 0.7905±0.0528 | 0.7969±0.0553 | 0.6749±0.0897 | 0.7299±0.0741 | 0.8470±0.0575 |
| | | testing | 0.6942±0.0434 | 0.6699±0.0866 | 0.5199±0.0645 | 0.5815±0.0534 | 0.7249±0.0500 |
| | SVMs | training | 0.7349±0.0106 | 0.6900±0.0163 | 0.6446±0.0183 | 0.6664±0.0141 | 0.7780±0.0178 |
| | | testing | 0.6865±0.0122 | 0.6583±0.0535 | 0.6093±0.0402 | 0.6305±0.0223 | 0.7296±0.0325 |
| *haberman* | AUC-LS-SVMs | training | 0.7720±0.0185 | 0.6766±0.0302 | 0.3432±0.0161 | 0.4552±0.0181 | 0.7822±0.0109 |
| | | testing | **0.7804±0.0437** | **0.5600±0.0787** | **0.3509±0.0609** | **0.4268±0.0522** | **0.7050±0.0333** |
| | LS-SVMs | traning | 0.8089±0.0653 | 0.7432±0.0991 | 0.4156±0.2058 | 0.5217±0.1777 | 0.8087±0.0752 |
| | | testing | 0.7185±0.0309 | 0.3923±0.1129 | 0.1883±0.0842 | 0.2452±0.0896 | 0.6700±0.0712 |
| | SVMs | training | 0.7453±0.0199 | N/A | 0 | N/A | 0.6056±0.0409 |
| | | testing | 0.7120±0.0462 | N/A | 0 | N/A | 0.5424±0.0653 |
| *german* | AUC-LS-SVMs | training | 0.9977±0.0024 | 0.9982±0.0016 | 0.9994±0.0013 | 0.9988±0.0012 | 0.9999±1.2025e-04 |
| | | testing | 0.9600±0.0156 | **0.9676±0.0090** | 0.9917±0.0090 | 0.9795±0.0081 | **0.7741±0.0931** |
| | LS-SVMs | training | 0.9783±0.0168 | 0.9782±0.0168 | 1±0 | 0.9889±0.0086 | 0.9834±0.0414 |
| | | testing | 0.9556±0.0113 | 0.9601±0.0116 | 0.9952±0.0080 | 0.9772±0.0059 | 0.7428±0.0746 |
| | SVMs | training | **0.9609±0.0066** | 0.9609±0.0066 | 1±0 | 0.9800±0.0034 | 0.9241±0.0238 |
| | | testing | 0.9649±0.0153 | 0.9649±0.0153 | **1±0** | **0.9821±0.0080** | 0.7525±0.0681 |
| *australian* | AUC-LS-SVMs | training | 0.8855±0.0089 | 0.8453±0.0153 | 0.9086±0.0134 | 0.8756±0.0066 | 0.9436±0.0089 |
| | | testing | **0.8694±0.0424** | **0.8294±0.0544** | **0.8978±0.0446** | **0.8614±0.0412** | **0.9375±0.0299** |
| | LS-SVMs | training | 0.8779±0.0266 | 0.8446±0.0471 | 0.8926±0.0338 | 0.8669±0.0263 | 0.9386±0.0210 |
| | | testing | 0.8448±0.0217 | 0.8135±0.0647 | 0.8587±0.0713 | 0.8310±0.0241 | 0.9185±0.0139 |
| | SVMs | training | 0.8796±0.0138 | 0.8283±0.0254 | 0.9155±0.0080 | 0.8696±0.0152 | 0.9435±0.0082 |
| | | testing | 0.8656±0.0274 | 0.8267±0.0520 | 0.8936±0.0414 | 0.8576±0.0328 | 0.9181±0.0300 |
| *svmguide1* | AUC-LS-SVMs | training | 0.9737±0.0039 | 0.9823±0.0047 | 0.9773±0.0062 | 0.9797±0.0030 | 0.9959±6.5113e-04 |
| | | testing | **0.9643±0.0073** | **0.9714±0.0107** | **0.9725±0.0172** | **0.9718±0.0068** | **0.9952±0.0022** |
| | LS-SVMs | training | 0.9852±0.0049 | 0.9864±0.0041 | 0.9906±0.0063 | 0.9885±0.0040 | 0.9987±4.8101e-04 |
| | | testing | 0.9522±0.0157 | 0.9565±0.0172 | 0.9699±0.0161 | 0.9630±0.0127 | 0.9890±0.0088 |
| | SVMs | training | 0.9417±0.0078 | 0.9561±0.0073 | 0.9526±0.0066 | 0.9543±0.0064 | 0.9864±0.0026 |
| | | testing | 0.9516±0.0160 | 0.9703±0.0127 | 0.9563±0.0182 | 0.9632±0.0125 | 0.9904±0.0050 |
| *svmguide3* | AUC-LS-SVMs | training | 0.9439±0.0035 | 0.9977±0.0052 | 0.7649±0.0217 | 0.8658±0.0129 | 0.9866±0.0024 |
| | | testing | **0.8128±0.0300** | 0.6399±0.1008 | **0.4977±0.0650** | **0.5593±0.0780** | **0.8061±0.0300** |
| | LS-SVMs | training | 0.8379±0.0267 | 0.8401±0.0524 | 0.3877±0.0987 | 0.5259±0.1025 | 0.8629±0.0357 |
| | | testing | 0.7936±0.0296 | 0.6923±0.1196 | 0.2640±0.0890 | 0.3723±0.0954 | 0.7404±0.0420 |
| | SVMs | training | 0.7992±0.0145 | 0.7933±0.0515 | 0.1619±0.0796 | 0.2613±0.1063 | 0.8001±0.0125 |
| | | testing | 0.7893±0.0301 | **0.7912±0.1641** | 0.1456±0.1057 | 0.2299±0.1351 | 0.7727±0.0271 |

Table 7: Evaluation on prostate cancer dataset - model (1)

| Methods | | accuracy | precision | recall | F1-score | AUC |
|---|---|---|---|---|---|---|
| AUC-LS-SVMs | training | 0.8831±0.0099 | 0.8175±0.0424 | 0.3606±0.0901 | 0.4966±0.0957 | 0.9042±0.0105 |
| | testing | **0.8831±0.0172** | **0.8453±0.0896** | 0.3898±0.0677 | **0.5328±0.0807** | **0.8936±0.0345** |
| LS-SVMs | training | 0.9319±0.0092 | 0.9302±0.0299 | 0.6285±0.0284 | 0.7499±0.0254 | 0.9555±0.0074 |
| | testing | 0.8768±0.0220 | 0.7523±0.1079 | **0.3968±0.0727** | 0.5145±0.0700 | 0.8443±0.0649 |
| SVMs | training | 0.8603±0.0195 | 0.8627±0.2106 | 0.2666±0.1974 | 0.3441±0.1901 | 0.9084±0.0235 |
| | testing | 0.8503±0.0372 | 0.7712±0.2804 | 0.2377±0.1950 | 0.2794±0.1444 | 0.8464±0.0791 |

Table 8: Evaluation on prostate cancer dataset - model (2)

| Methods | | accuracy | precision | recall | F1-score | AUC |
|---|---|---|---|---|---|---|
| AUC-LS-SVMs | training | 0.9684±0.0047 | 0.9487±0.0595 | 0.6965±0.0990 | 0.7970±0.0410 | 0.9779±0.0062 |
| | testing | 0.9273±0.0188 | 0.7225±0.2410 | **0.5233±0.1119** | **0.5834±0.1161** | **0.8679±0.0574** |
| LS-SVMs | training | 0.9376±0.0062 | 0.9319±0.0436 | 0.4939±0.0452 | 0.6393±0.0409 | 0.9345±0.0147 |
| | testing | 0.9240±0.0065 | 0.4924±0.2208 | 0.4207±0.0636 | 0.5467±0.0587 | 0.8525±0.0226 |
| SVMs | training | 0.9501±0.0063 | 0.9092±0.0195 | 0.4539±0.0665 | 0.5915±0.0533 | 0.9186±0.0194 |
| | testing | **0.9377±0.0116** | **0.8083±0.0550** | 0.3427±0.1262 | 0.4723±0.1280 | 0.8403±0.0732 |

Table 9: Baseline characteristics of the dataset for model (1)

| | Value | Percentage |
|---|---|---|
| Total number of patients | 3435 | |
| Number and percentage of patients with PSA level (ng $ml^{-1}$) | | |
| <4 | 339 | 9.87 |
| 4-10 | 1799 | 52.37 |
| 10.1-20 | 713 | 20.76 |
| 20.1-50 | 295 | 8.59 |
| >50 | 289 | 8.41 |
| PSA level (ng $ml^{-1}$) | 46.66±349.72 | |
| Age(year, mean±s.d.) | 68±8 | |
| Estimated prostate volume on TRUS (ml, mean±s.d.) | 51.23±26.31 | |
| TRUS finding (number of patients) | | |
| Normal | 3058 | 89.02 |
| Abnormal | 377 | 10.98 |
| Outcome of biopsy | | |
| non-cancer or insignificant cancer | 2939 | |
| significant cancer | 496 | |

Table 10: Baseline characteristics of the dataset for model (2)

| | Value | Percentage |
|---|---|---|
| Total number of patients | 1973 | |
| Number and percentage of patients with PSA level (ng $ml^{-1}$) | | |
| <4 | 114 | 5.78 |
| 4-10 | 1261 | 63.91 |
| 10.1-20 | 377 | 19.11 |
| 20.1-50 | 141 | 7.15 |
| >50 | 80 | 4.05 |
| PSA level (ng $ml^{-1}$) | 23.24±121.08 | |
| Age(year, mean±s.d.) | 67±7 | |
| Estimated prostate volume on DRE (ml, mean±s.d.) | 45.88±16.59 | |
| DRE findings | | |
| normal | 1717 | 87.02 |
| abnormal | 256 | 12.98 |
| Outcome of biopsy | | |
| non-cancer or insignificant cancer | 1751 | |
| significant cancer | 222 | |

## Appendix

Eq. (10) can be reformulated as

$$\begin{aligned}\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}+\frac{\gamma}{2}\sum_{i=1}^{N}\xi_i^2+ \\ & \frac{C}{2}\sum_{k\in N_+}\sum_{l\in N_-}\frac{\big(\lambda-\boldsymbol{w}^T(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l))\big)^2}{n_+n_-} \\ \text{s.t} \quad & y_i=\boldsymbol{w}^T\varphi(\boldsymbol{x}_i)+b+\xi_i, i=1,2,\cdots,N \\ & (N=n^++n^-)\end{aligned} \tag{23}$$

To derive the dual problem by constructing the Lagrangian, we formulate the Lagrangian $J$ for Eq. (23)

$$\begin{aligned}J=&\frac{1}{2}\boldsymbol{w}^2+\frac{C}{2}\sum_{k\in N_+}\sum_{l\in N_-}\frac{\big(\lambda-\boldsymbol{w}^T(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l))\big)^2}{n_+n_-} \\ &+\frac{\gamma}{2}\sum_{i=1}^{N}\xi_i^2+\sum_{i=1}^{N}\alpha_i(y_i-\boldsymbol{w}^T\varphi(\boldsymbol{x}_i)-b-\xi_i)\end{aligned} \tag{24}$$

where $\boldsymbol{\alpha}_i=(\alpha_1,\alpha_2,...,\alpha_N)$ is the vector of Lagrangian multipliers. The conditions for optimality are given by

$$\begin{aligned}&\frac{\partial J}{\partial \boldsymbol{w}}=0\Rightarrow \boldsymbol{w}+C\sum_{k\in N^+}\sum_{l\in N^-}\frac{\Big(\lambda-\boldsymbol{w}^T\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)\Big)}{n^+n^-} \\ &\Big(-\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)\Big)-\sum_{i=1}^{N}\alpha_i\boldsymbol{x}_i=0 \\ &\Rightarrow \boldsymbol{w}+\frac{C}{n^+n^-}\sum_{k\in N^+}\sum_{l\in N^-}\Big(\varphi(\boldsymbol{x}_l)-\varphi(\boldsymbol{x}_k)\Big) \\ &\Big(\lambda-\boldsymbol{w}^T\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)\Big)-\sum_{i=1}^{N}\alpha_i\varphi(\boldsymbol{x}_i)=0\end{aligned} \tag{25}$$

Since $\boldsymbol{w}^T\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)$ is scalar, $\boldsymbol{w}^T\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)=\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)^T\boldsymbol{w}$. We can further write Eq. (25) into

$$\begin{aligned}&\frac{\partial J}{\partial \boldsymbol{w}}=0\Rightarrow \boldsymbol{w}+\frac{\lambda C}{n^+n^-}\sum_{k\in N^+}\sum_{l\in N^-}\big(\varphi(\boldsymbol{x}_l)-\varphi(\boldsymbol{x}_k)\big) \\ &+\frac{C}{n^+n^-}\sum_{k\in N^+}\sum_{l\in N^-}\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)\big(\varphi(\boldsymbol{x}_k) \\ &-\varphi(\boldsymbol{x}_l)\big)^T\boldsymbol{w}-\sum_{i=1}^{N}\alpha_i\varphi(\boldsymbol{x}_i)=0 \\ &\Rightarrow \boldsymbol{w}=\mathbf{H}\Big(\sum_{i=1}^{N}\alpha_i\varphi(\boldsymbol{x}_i)+\frac{\lambda C}{n_+n_-}\sum_{k\in N+}\sum_{l\in N-} \\ &\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)\Big)\end{aligned} \tag{26}$$

where $\mathbf{H}=\Big[\boldsymbol{I}+\frac{C}{n^+n^-}\sum_{k\in N+}\sum_{l\in N-}\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)^T\Big]^{-1}$, $\boldsymbol{I}$ is the $N\times N$ identity matrix and $\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)^T$ is an $N\times N$ matrix.

$$\frac{\partial J}{\partial b}=0 \quad \Rightarrow\sum_{i=1}^{N}\alpha_i=0 \tag{27}$$

$$\frac{\partial J}{\partial \xi_i}=0 \quad \Rightarrow\alpha_i=\gamma\xi_i \tag{28}$$

$$\frac{\partial J}{\partial \alpha_i}=0 \quad \Rightarrow y_i=\boldsymbol{w}^T\varphi(\boldsymbol{x}_i)+b+\xi_i \tag{29}$$

According to Sherman-Morrison-Woodbury formula [33], given an invertible (nonsingular) matrix $\mathbf{A}$ and column vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, assuming $1+\boldsymbol{v}^T\mathbf{A}^{-1}\boldsymbol{u}\neq 0$, we have

$$(\mathbf{A}+\boldsymbol{u}\boldsymbol{v}^T)^{-1}=\mathbf{A}^{-1}-\frac{\mathbf{A}^{-1}\boldsymbol{u}\boldsymbol{v}^T\mathbf{A}^{-1}}{1+\boldsymbol{v}^T\mathbf{A}^{-1}\boldsymbol{u}} \tag{30}$$

In particular if $\mathbf{A}=\boldsymbol{I}$, we immediately have $(\boldsymbol{I}+\boldsymbol{u}\boldsymbol{v}^T)^{-1}=\boldsymbol{I}-\frac{\boldsymbol{u}\boldsymbol{v}^T}{1+\boldsymbol{v}^T\boldsymbol{u}}$. By applying this formula to $\mathbf{H}$, we can rewrite $\mathbf{H}$ into

$$\begin{aligned}\mathbf{H}=&\boldsymbol{I}-\frac{C}{n^+n^-}\sum_{k\in N^+}\sum_{l\in N^-} \\ &\frac{\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)^T}{\Big[1+\frac{C}{n^+n^-}\sum_{k\in N^+}\sum_{l\in N^-}\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)^T\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)\Big]} \\ =&\boldsymbol{I}- \\ &\frac{\sum_{k\in N^+}\sum_{j\in N^-}\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)^T}{\frac{n^+n^-}{C}+\sum_{k\in N^+}\sum_{l\in N^-}\big(k(\boldsymbol{x}_k,\boldsymbol{x}_k)+k(\boldsymbol{x}_l,\boldsymbol{x}_l)-2k(\boldsymbol{x}_k,\boldsymbol{x}_l)\big)}\end{aligned} \tag{31}$$

We notice that the denominator in Eq. (31) is a scalar. If we use $M$ to represent it, Eq. (31) can be simplified into

$$\mathbf{H}=\boldsymbol{I}-\frac{\sum_{k\in N^+}\sum_{l\in N^-}\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)^T}{M} \tag{32}$$

and accordingly Eq. (26) can be simplified into

$$\begin{aligned}\boldsymbol{w}=&\Big(\boldsymbol{I}-\frac{1}{M}\sum_{k\in N^+}\sum_{l\in N^-}\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)^T\Big) \\ &\Big(\sum_{i=1}^{N}\alpha_i\varphi(\boldsymbol{x}_k)+\frac{\lambda C}{n^+n^-}\sum_{k\in N^+}\sum_{l\in N^-}\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)\Big)\end{aligned} \tag{33}$$

By eliminating $\boldsymbol{w}$ and $\xi_i$, we can get the following solution

$$
\begin{aligned}
y_i &= \varphi^T(\boldsymbol{x}_i)\Big(\boldsymbol{I} - \frac{1}{M}\sum_{k\in N^+}\sum_{l\in N^-}\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)^T\Big) \\
&\Big(\sum_{i=1}^{N}\alpha_i\varphi(\boldsymbol{x}_i) + \frac{\lambda C}{n^+n^-}\sum_{k\in N^+}\sum_{l\in N^-}\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)\Big) + b + \frac{\alpha_i}{\gamma} \\
&= \Big(\varphi^T(\boldsymbol{x}_i) - \frac{1}{M}\sum_{k\in N^+}\sum_{l\in N^-}\big(k(\boldsymbol{x}_i,\boldsymbol{x}_k)-k(\boldsymbol{x}_k,\boldsymbol{x}_l)\big)\big(\varphi(\boldsymbol{x}_k)- \\
&\varphi(\boldsymbol{x}_l)\big)^T\Big)\Big(\sum_{i=1}^{N}\alpha_i\varphi(\boldsymbol{x}_i) + \frac{\lambda C}{n^+n^-}\sum_{k\in N^+}\sum_{l\in N^-}\big(\varphi(\boldsymbol{x}_k)-\varphi(\boldsymbol{x}_l)\big)\Big) \\
&+ b + \frac{\alpha_i}{\gamma} \\
&= \sum_{k=1}^{N}\alpha_k\Big[k(\boldsymbol{x}_i,\boldsymbol{x}_k) - \frac{1}{M}\sum_{p\in N^+}\sum_{l\in N^-}\big(k(\boldsymbol{x}_i,\boldsymbol{x}_p)-k(\boldsymbol{x}_i,\boldsymbol{x}_l)\big) \\
&\big(k(\boldsymbol{x}_p,\boldsymbol{x}_k)-k(\boldsymbol{x}_l,\boldsymbol{x}_k)\big)\Big] + \frac{\lambda C}{n^+n^-}\sum_{k\in N^+}\sum_{l\in N^-}\Big\{\big(k(\boldsymbol{x}_i,\boldsymbol{x}_k)- \\
&k(\boldsymbol{x}_i,\boldsymbol{x}_l)\big) - \frac{1}{M}\sum_{p\in N^+}\sum_{q\in N^-}\big(k(\boldsymbol{x}_i,\boldsymbol{x}_p)-k(\boldsymbol{x}_i,\boldsymbol{x}_q)\big)\sum_{k\in N^+}\sum_{l\in N^-} \\
&\big(k(\boldsymbol{x}_p,\boldsymbol{x}_k)-k(\boldsymbol{x}_p,\boldsymbol{x}_l)-k(\boldsymbol{x}_q,\boldsymbol{x}_k)+k(\boldsymbol{x}_q,\boldsymbol{x}_l)\big)\Big\} + b + \frac{\alpha_i}{\gamma}
\end{aligned}
\tag{34}
$$

We denote $k(\boldsymbol{x}_i,\boldsymbol{x}_k) - \frac{1}{M}\sum_{p\in N^+}\sum_{l\in N^-}\big(k(\boldsymbol{x}_i,\boldsymbol{x}_p) - k(\boldsymbol{x}_i,\boldsymbol{x}_l)\big)\big(k(\boldsymbol{x}_p,\boldsymbol{x}_k)-k(\boldsymbol{x}_l,\boldsymbol{x}_k)\big)$ as $\tilde{k}(\boldsymbol{x}_i,\boldsymbol{x}_k)$, and $\frac{C}{n^+n^-}\sum_{k\in N^+}\sum_{l\in N^-}\Big\{\big(k(\boldsymbol{x}_i,\boldsymbol{x}_k)-k(\boldsymbol{x}_i,\boldsymbol{x}_l)\big) - \frac{1}{M}\sum_{p\in N^+}\sum_{q\in N^-}\big(k(\boldsymbol{x}_i,\boldsymbol{x}_p)-k(\boldsymbol{x}_i,\boldsymbol{x}_q)\big)\sum_{k\in N^+}\sum_{l\in N^-}\big(k(\boldsymbol{x}_p,\boldsymbol{x}_k) - k(\boldsymbol{x}_p,\boldsymbol{x}_l) - k(\boldsymbol{x}_q,\boldsymbol{x}_k) + k(\boldsymbol{x}_q,\boldsymbol{x}_l)\big)\Big\}$ as $f(\boldsymbol{x}_i)$, therefore we can rewrite Eq. (34) into

$$
y_i = \sum_{k=1}^{N}\alpha_k\tilde{k}(\boldsymbol{x}_i,\boldsymbol{x}_k) + \lambda f(\boldsymbol{x}_i) + b + \frac{\alpha_i}{\gamma} \tag{35}
$$

We can further write the above linear equation in the matrix form

$$
\begin{bmatrix}\tilde{\mathbf{K}} + \frac{\boldsymbol{I}}{\gamma} & \mathbf{1} \\ \mathbf{1}^T & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha} \\ b\end{bmatrix} = \begin{bmatrix}\boldsymbol{y} - \lambda\boldsymbol{f} \\ 0\end{bmatrix} \tag{36}
$$

where $\boldsymbol{y} = [y_1;\cdots;y_N]^T$, $\mathbf{1} = [1;\cdots;1]$, $\boldsymbol{f} = [f(\boldsymbol{x}_1);\cdots;f(\boldsymbol{x}_N)]^T$, and $\tilde{\mathbf{K}} = (\tilde{k}(\boldsymbol{x}_i,\boldsymbol{x}_k))_{N\times N}$.

## Acknowledgments

The work was supported by the Innovation and Technology Commission of the Government of the Hong Kong SAR under the ITF-MRP project (MRP/015/18), the Australian Research Council (ARC) under Discovery Grant DP170101632 and G. Wang is supported by Murdoch New Staff Startup Grant (SEIT NSSG).

## References

1. Cancer stat facts: Prostate cancer. https://seer.cancer.gov/statfacts/html/prost.html. Accessed: 2018-04-30
2. From development to use in clinical practice - ERSPC prostate cancer risk calculator. http://www.prostatecancer-riskcalculator.com/from-development-to-use-in-clinical-practice-erspc-prostate-cancer-risk-calculator
3. LIBSVM Data: Classification (Binary Class). https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/binary.html
4. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/datasets.html
5. Optimising area under the ROC curve using gradient descent. In: Proceedings of the Twenty-first International Conference on Machine Learning, p. 49. ACM (2004)
6. Ablin, R., Pfeiffer, L., Gonder, M., Soanes, W.: Precipitating antibody in the sera of patients treated cryosurgically for carcinoma of the prostate. Experimental Medicine and Surgery **27**(4), 406–410 (1968)
7. Artan, Y., Haider, M.A., Langer, D.L., Van der Kwast, T.H., Evans, A.J., Yang, Y., Wernick, M.N., Trachtenberg, J., Yetik, I.S.: Prostate cancer localization with multispectral mri using cost-sensitive support vector machines and conditional random fields. IEEE Transactions on Image Processing **19**(9), 2444–2455 (2010)
8. Brefeld, U., Scheffer, T.: AUC maximizing support vector learning. In: Proceedings of the International Conference on Machine Learning (ICML) 2005 workshop on ROC Analysis in Machine Learning (2005)
9. Calders, T., Jaroszewicz, S.: Efficient AUC optimization for classification. In: European Conference on Principles of Data Mining and Knowledge Discovery, pp. 42–53. Springer (2007)
10. Catalona, W., Hudson, M., Scardino, P., Richie, J., Ahmann, F., Flanigan, R., DeKernion, J., Ratliff, T., Kavoussi, L., Dalkin, B.: Selection of optimal prostate specific antigen cutoffs for early detection of prostate cancer: receiver operating characteristic curves. The Journal of Urology **152**(6 Pt 1), 2037–2042 (1994)
11. Catalona, W., Richie, J., Ahmann, F., Hudson, M., Scardino, P., Flanigan, R., Dekernion, J., Ratliff, T., Kavoussi, L., Dalkin, B.: Comparison of digital rectal examination and serum prostate specific antigen in the early detection of prostate cancer: results of a multicenter clinical trial of 6,630 men. The Journal of Urology **151**(5), 1283–1290 (1994)
12. Cawley, G.C.: Leave-one-out cross-validation based model selection criteria for weighted ls-svms. In: The 2006 IEEE International Joint Conference on Neural Network Proceedings, pp. 1661–1668. IEEE (2006)
13. Chang, C.C., Lin, C.J.: LIBSVM: a library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology (TIST) **2**(3), 27 (2011)
14. Chawla, N.V., Japkowicz, N., Kotcz, A.: Special issue on learning from imbalanced data sets. ACM SIGKDD Explorations Newsletter **6**(1), 1–6 (2004)
15. Çınar, M., Engin, M., Engin, E.Z., Ateşçi, Y.Z.: Early prostate cancer diagnosis by using artificial neural networks and support vector machines. Expert Systems with Applications **36**(3), 6357–6361 (2009)
16. Cortes, C., Mohri, M.: AUC optimization vs. error rate minimization. In: Advances in Neural Information Processing Systems, pp. 313–320 (2004)

17. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20**(3), 273–297 (1995)
18. Elkan, C.: The foundations of cost-sensitive learning. In: International Joint Conference on Artificial Intelligence, vol. 17, pp. 973–978. Lawrence Erlbaum Associates Ltd (2001)
19. Gao, W., Jin, R., Zhu, S., Zhou, Z.H.: One-pass AUC optimization. In: International Conference on Machine Learning, pp. 906–914 (2013)
20. Gao, W., Zhou, Z.H.: On the consistency of AUC pairwise optimization. In: International Joint Conference on Artificial Intelligence (IJCAI), pp. 939–945 (2015)
21. Ghazikhani, A., Monsefi, R., Yazdi, H.S.: Online neural network model for non-stationary and imbalanced data stream classification. International Journal of Machine Learning and Cybernetics **5**(1), 51–62 (2014)
22. Hanley, J.A., McNeil, B.J.: The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology **143**(1), 29–36 (1982)
23. Holst, A., et al.: Efficient AUC maximization with regularized least-squares. In: Tenth Scandinavian Conference on Artificial Intelligence: SCAI 2008, vol. 173, p. 12. IOS Press (2008)
24. Joachims, T.: A support vector method for multivariate performance measures. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 377–384. ACM (2005)
25. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence **5**(4), 221–232 (2016)
26. Lee, W., Jun, C.H., Lee, J.S.: Instance categorization by support vector machines to adjust weights in AdaBoost for imbalanced data classification. Information Sciences **381**, 92–103 (2017)
27. Li, S., Zhang, Y., Xu, J., Li, L., Zeng, Q., Lin, L., Guo, Z., Liu, Z., Xiong, H., Liu, S.: Noninvasive prostate cancer screening based on serum surface-enhanced raman spectroscopy and support vector machine. Applied Physics Letters **105**(9), 091104 (2014)
28. Liu, Y.: Active learning with support vector machine applied to gene expression data for cancer classification. Journal of Chemical Information and Computer Sciences **44**(6), 1936–1941 (2004)
29. Mao, W., Wang, J., Xue, Z.: An ELM-based model with sparse-weighting strategy for sequential data imbalance problem. International Journal of Machine Learning and Cybernetics **8**(4), 1333–1345 (2017)
30. Nadji, M., Tabei, S.Z., Castro, A., Chu, T.M., Murphy, G.P., Wang, M.C., Morales, A.R.: Prostatic-specific antigen: An immunohistologic marker for prostatic neoplasms. Cancer **48**(5), 1229–1232 (1981)
31. Rakotomamonjy, A.: Optimizing area under ROC curve with SVMs. In: ROCAI, pp. 71–80 (2004)
32. Rezvani, S., Wang, X., Pourpanah, F.: Intuitionistic fuzzy twin support vector machines. IEEE Transactions on Fuzzy Systems **27**(11), 2140–2151 (2019)
33. Riedel, K.S.: A Sherman–Morrison–Woodbury identity for rank augmenting matrices with application to centering. SIAM Journal on Matrix Analysis and Applications **13**(2), 659–662 (1992)
34. Suykens, J., Van Gestel, T., De Brabanter, J., De Moor, B., Vandewalle, J.: Least Squares Support Vector Machine Classifiers. World Scientific, Singapore (2002)
35. Vapnik, V.N.: An overview of statistical learning theory. IEEE Transactions on Neural Networks **10**(5), 988–999 (1999)
36. Wang, G., Lu, J., Choi, K.S., Zhang, G.: A transfer-based additive ls-svm classifier for handling missing data. IEEE Transactions on Cybernetics (2018)
37. Wang, G., Zhang, G., Choi, K.S., Lu, J.: Deep additive least squares support vector machines for classification with model transfer. IEEE Transactions on Systems, Man, and Cybernetics: Systems (2017)
38. Ye, J., Xiong, T.: SVM versus least squares SVM. In: Artificial Intelligence and Statistics, pp. 644–651 (2007)
39. Ying, Y., Wen, L., Lyu, S.: Stochastic online AUC maximization. In: Advances in Neural Information Processing Systems, pp. 451–459 (2016)
40. Zhang, C., Zhou, Y., Guo, J., Wang, G., Wang, X.: Research on classification method of high-dimensional class-imbalanced datasets based on SVM. International Journal of Machine Learning and Cybernetics pp. 1–14 (2018). URL https://doi.org/10.1007/s13042-018-0853-2
41. Zhang, K., Kwok, J.T.: Simplifying mixture models through function approximation. IEEE Transactions on Neural Networks **21**(4), 644–658 (2010)
42. Zhao, P., Hoi, S.C., Jin, R., YANG, T.: Online AUC maximization. In: Proceedings of the 28th International Conference on Machine Learning ICML. International Machine Learning Society (2011)
43. Zhou, Z.H., Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering **18**(1), 63–77 (2006)
44. Zhu, Z., Wang, Z., Li, D., Du, W.: Multiple empirical kernel learning with majority projection for imbalanced problems. Applied Soft Computing **76**, 221–236 (2019)