Analysis of presence-only data via semi-supervised learning approaches

Junhui Wang
Department of Mathematics, Statistics,
and Computer Science
University of Illinois at Chicago
Chicago, IL 60607

Yixin Fang
Division of Biostatistics
School of Medicine
New York University
New York, NY 10016

Abstract

Presence-only data occur in classification, which consist of a sample of observations from presence class and a large number of background observations with unknown presence/absence. Since absence data are generally unavailable, conventional semi-supervised learning approaches are no longer appropriate as they tend to degenerate and assign all observations to presence class. In this article, we propose a generalized class balance constraint, which can be equipped with semi-supervised learning approaches to prevent them from degeneration. Furthermore, to circumvent the difficulty of model tuning with presence-only data, a selection criterion based on classification stability is developed, which measures the robustness of any given classification algorithm against the sampling randomness. The effectiveness of the proposed approach is demonstrated through a variety of simulated examples, along with an application to gene function prediction.

Key words: Cross validation, Functional genomics, Stability, Support vector machine, Tuning

1 Introduction

Presence-only data, also known as positive and unlabeled data, consist of a sample of observations from presence (or positive) class and a large number of background (or unlabeled) observations with unknown class labels. It has been commonly encountered in many real applications, ranging from information technology and computer engineering (Liu et al., 2003), ecological modeling (Ward et al., 2009), to biomedical sciences (Zhao et al., 2008). In biomedical sciences, gene function prediction is one of the primary goals in understanding genomics. Thanks to the rapid advance in high-throughput biotechnologies, a large amount of gene expression profiles have been obtained. However, based on the available expression profiles, annotating genes with biological function classes is still labor intensive and time consuming. In general, it is relatively easy to identify genes annotated with certain function of interest, while it is much harder to find which genes do not have the function (Zhao et al., 2008). In such situations, the primary goal is to leverage the background data to enhance predictive performance of classification.

To incorporate the background data into analysis, many methods have been proposed in the literature of statistics and machine learning. Among others, the naive method (Keating and Cherry, 2004) constructs a supervised classification by treating all background data as observations from the absence class; the iterative method (Liu et al., 2003) iteratively expands the absence sample by adding background data that are most likely from the absence class; and an expectation-maximization (EM) algorithm (Ward et al., 2009) treats the class labels of the background data as missing and fits an underlying presence-absence logistic regression model by assuming some prior knowledge on the presence frequency.

Note that presence-only data can naturally fit into the framework of Semi-Supervised Learning (SSL), which is a special type of classification problem with only a small labeled sample and a large unlabeled background sample. However, analysis of presence-only data via

SSL approaches seems rare in the literature, probably due to the following two reasons. First, most existing SSL approaches, such as the Transductive Support Vector Machine (TSVM; Vapnik, 1998; Chapelle and Zien, 2005; Wang, Shen and Pan, 2007) and the efficient SSL (Wang, Shen and Pan, 2009), require that the labeled sample should consist of observations from both the presence and the absence classes. Clearly, this requirement is violated in presence-only data as the labeled sample contains only observations from the presence class, and hence that directly applying the existing SSL approaches to presence-only data yields degenerate classification function assigning all observations to the presence class. Second, most existing SSL approaches rely on tuning parameters to balance the tradeoff between the labeled and the unlabeled data. Cross validation (CV) is popularly employed to select the optimal tuning parameters in terms of the classification accuracy. However, the conventional CV procedure is no longer appropriate for presence-only data, where no observation from the absence class is available and the estimated classification accuracy based on CV may become unreliable.

The main contribution of this article is to overcome two encountered issues when analyzing presence-only data with SSL approaches. First, it extends the existing SSL approaches by enforcing a generalized class balance constraint, which restricts the candidate classification function space to more informative regions as opposed to those degenerate ones assigning all observations to one single class. The proposed class balance constraint can be adapted to most margin-based SSL approaches including the TSVM and the efficient SSL. Second, to overcome the difficulty of lacking absence data in model tuning, a novel tuning criterion is developed based on classification stability, which measures the robustness of any given classification algorithm against the sampling randomness. The classification stability does not require the class label of the absence data, and thus can be accurately estimated by using both presence and absence data. The effectiveness of the proposed approach is demonstrated in numerical experiments with both simulated examples and a real application in

gene function prediction, which has been central to biomedical research in recent years.

The rest of this article is organized as follows. Section 2 introduces the presence-only data analysis and SSL. Section 3 proposes the generalized class balance constraint and extends the TSVM and the efficient SSL to presence-only data analysis. Section 4 develops the model tuning criterion, as well as its estimation scheme and local asymptotic consistency. Section 5 presents some numerical examples, together with an application to gene function prediction. Section 6 contains a summary, and the Appendix is devoted to technical proofs.

2 Presence-only data analysis and SSL

A typical presence-only dataset consists of a presence sample $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n_l}$ with $\mathbf{x}_i \in \mathbf{R}^d$ and $y_i \equiv 1$ and a background sample $\{\mathbf{x}_j\}_{j=n_l+1}^{n}$ with $n_u = n - n_l$. Let $z_i = 1$ if the observation is in the presence sample and $z_i = -1$ if the observation is in the background sample. Clearly, $z_i = 1$ indicates that $y_i = 1$ while $z_i = -1$ provides no information on whether $y_i = -1$ or 1. Naturally presence-only data analysis can be formulated as a special SSL problem by treating the presence data as labeled and the background data as unlabeled.

SSL has attracted enormous attention from both statistics and machine learning communities; for example, a recent survey paper on SSL (Zhu, 2005) has cited over 100 references. In principle, the unlabeled data can improve the classification performance given a strong match of data structure with some model assumptions on the connection between the labeled and the unlabeled data, while weak match of data structure with model assumptions offers no help at all or even deteriorates the performance. Various assumptions have been proposed in the literature, leading to different approaches. Here we just briefly review three popular assumptions and refer to Zhu (2005) for a much more extensive literature review on SSL. Smoothness assumption assumes that neighboring instances tend to share the same class label, which often requires additional assumptions on defining the neighborhood as well as the

relationship between neighborhood and labeling, such as in the Gaussian random field (Zhu, Ghahramani and Lafferty, 2003). Clustering assumption (Chapelle and Zien, 2005) assumes that the classification decision boundary should be close to the clustering boundary, such as in the margin-based SSL approaches, including the TSVM and the efficient SSL. Manifold assumption (Belkin and Niyogi, 2004) is similar to clustering assumption, and assumes that the data and classification decision boundary reside on a low-dimensional manifold estimated based on the unlabeled data, which leads to a manifold regularized SSL formulation (Belkin et al., 2006; Tian et al., 2012).

In this article, we focus on the margin-based SSL under the clustering assumption. In specific, the margin-based SSL approaches estimate the classification function $f(\mathbf{x})$ by solving

$$\min_{f \in \mathcal{F}} \quad C_1 \sum_{i=1}^{n_l} L(y_i f(\mathbf{x}_i)) + C_2 \sum_{j=n_l+1}^{n} U(f(\mathbf{x}_j)) + J(f),$$

where $L(yf(\mathbf{x}))$ and $U(f(\mathbf{x}))$ are loss functions for labeled and unlabeled data respectively, $J(f)$ is a penalty term on complexity of f, and $\hat{\phi}(\mathbf{x}) = \text{sign}(\hat{f}(\mathbf{x}))$ is the classification decision function. The $L(yf(\mathbf{x}))$ loss can be any margin-based loss function, such as the hinge loss $(1 - yf(\mathbf{x}))_+$, and the $U(f(\mathbf{x}))$ loss connects the classification function f and the clustering structure of the unlabeled data. For instances, $U(f(\mathbf{x})) = (1 - |f(\mathbf{x})|)_+$ leads to the TSVM, and $U(f(\mathbf{x})) = \hat{p}(\mathbf{x})L(f(\mathbf{x})) + (1 - \hat{p}(\mathbf{x}))L(-f(\mathbf{x}))$ leads to the efficient SSL, where $\hat{p}(\mathbf{x})$ is an estimate of $p(\mathbf{x}) = P(Y = 1|X = \mathbf{x})$.

Furthermore, the minimization of the margin-based SSL formulation is often solved under a class balance constraint (Joachims, 1999; Chapelle, Sindhwani and Keerthi, 2008) that

$$\sum_{i=1}^{n} \text{sign}(f(\mathbf{x}_i)) = (2r - 1)n. \tag{1}$$

This constraint enforces that a pre-specified proportion, r, of the training data should be

assigned to the positive class, avoiding a degenerated classification function. Since (1) is nonlinear and difficult for implementation, it is often relaxed to $\sum_{i=1}^{n} f(\mathbf{x}_i) = 2r - 1$, where the pre-specified proportion r is estimated by the proportion of the presence and absence data in the labeled sample. However, in presence-only data, estimation of r becomes infeasible due to the lack of absence data (Ward et al., 2009). Without accurate knowledge of the proportion of presence sample, (1) can be even harmful in that it may force the SSL approaches to sacrifice their classification performance in order to achieve the restrictive constraint with misspecified proportion.

3 SSL under a generalized class balance constraint

This section proposes a generalized class balance constraint to prevent the classification function from degeneration, which relaxes the restrictive equality constraint in (1) to an inequality constraint,

$$\left| \sum_{i=1}^{n} \text{sign}(f(\mathbf{x}_i)) \right| \leq D, \tag{2}$$

where D is a pre-specified constant controlling the balance between positive and negative predictions of f. Since (2) is a quantile-type constraint and infeasible for implementation, it can be further relaxed to

$$-D \leq \sum_{i=1}^{n} f(\mathbf{x}_i) \leq D, \tag{3}$$

which is linear and easy to be implemented.

In (3), $D \geq n$ admits $f(x) \equiv 1$ and can not prevent degeneration, whereas a smaller value of D can guarantee a nondegenerated solution. More importantly, the classification performance of the SSL approaches is relatively insensitive to the value of D. Figure 1 displays the classification performance of the TSVM with the equality constraint $\sum_{i=1}^{n} f(\mathbf{x}_i) = D$ and the inequality constraints (3) as functions of the tuning parameter D.

Insert Figure 1 about here

As showed in Figure 1, the testing error of the TSVM with the equality constraint appears to much more variable as D changes, while the testing error of the TSVM with the inequality constraint stays the same for a long range of D and then shoot up to the degenerate case when D is too large. Consequently, to achieve the optimal classification performance, the TSVM with the equality constraint has to search for the appropriate tuning parameters (D, C_1, C_2), but the TSVM with the inequality constraint may only need to tune two parameters (C_1, C_2) under a pre-specified D without sacrificing the classification performance. In all the numerical experiments, $D = n_u - n_l$ appears to work reasonably well as it assures that the background sample are mixed with both presence and absence observations. In practice, D may be set based on a rough estimate of the presence sample size if prior knowledge is available.

In the sequel, we will focus on two margin-based presence-only SSL approaches, PO-TSVM and PO-ESSL. The PO-TSVM equips the generalized class balance constraint to the orignal TSVM formulation, which seeks the largest possible separation of both the labeled and the unlabeled data. Specifically, the PO-TSVM classification function $\hat{f}(\mathbf{x})$ is obtained by solving

$$\begin{aligned} \min_{f\in\mathcal{F}} \quad & C_1 \sum_{i=1}^{n_l} (1 - y_i f(\mathbf{x}_i))_+ + C_2 \sum_{j=n_l+1}^{n} (1 - |f(\mathbf{x}_j)|)_+ + \|f\|_{\mathcal{F}}^2, \\ \text{subject to} \quad & -D \le \sum_{i=1}^{n} f(\mathbf{x}_i) \le D, \end{aligned} \tag{4}$$

where $\|\cdot\|_{\mathcal{F}}^2$ is the reproducing kernel Hilbert space (RKHS) norm. The PO-ESSL is based on a novel $U(f(\mathbf{x}))$ for the unlabeled data as in Wang et al. (2009), which seeks efficient extraction of information from the unlabeled data for estimating the optimal classification

function. Specifically, the PO-ESSL estimates $f(\mathbf{x})$ by solving

$$\begin{aligned} \min_{f\in\mathcal{F}} \quad & C_1\sum_{i=1}^{n_l}(1-y_i f(\mathbf{x}_i))_+ + C_2\sum_{j=n_l+1}^{n}\hat{U}(f(\mathbf{x}_j)) + \|f\|_{\mathcal{F}}^2, \\ \text{subject to} \quad & -D \le \sum_{i=1}^{n} f(\mathbf{x}_i) \le D, \end{aligned} \tag{5}$$

where $\hat{U}(f(\mathbf{x})) = \hat{p}(\mathbf{x})L(f(\mathbf{x})) + (1-\hat{p}(\mathbf{x}))L(-f(\mathbf{x}))$ is an estimate of the efficient margin loss for the unlabeled data, $U(f(\mathbf{x})) = p(\mathbf{x})L(f(\mathbf{x})) + (1-p(\mathbf{x}))L(-f(\mathbf{x}))$.

To solve (4), a difference convex algorithm can be employed as in Wang et al. (2007). It decomposes the non-convex cost function in (4) as a difference of two convex functions, approximates the second convex function by it gradient, and solves (4) by sequential quadratic programming (QP). Note that the class balance constraint does not increase the computational cost at all, as it only adds two linear inequality constraints in the QP routine. The computation cost can be further reduced when the QP routine is replaced by a more efficient gradient descent method (Guan et al., 2012).

To solve (5), an iterative scheme can be implemented. First, an initial $\hat{f}(\mathbf{x})$ is constructed. Second, given $\text{sign}(\hat{f}(\mathbf{x}))$, $\hat{p}(\mathbf{x})$ is obtained through the procedure in Wang et al. (2008). Third, $\hat{f}(\mathbf{x})$ is updated by solving (5) with $\hat{p}(\mathbf{x})$ fixed through a QP routine. The last two steps can be iterated until convergence. Note that the first step can be initialized by any presence-only approach, including the naive method, the iterative method and the PO-TSVM, and different initialization methods may yield different final estimated $\hat{f}(\mathbf{x})$. More interestingly, through the iterative optimization scheme, the PO-ESSL can be thought of as a reminiscent of the iterative method, but it updates the reliable absence sample more appropriately, where all the background data are included in the reliable absence sample but their contributions are determined by the magnitude of the corresponding $\hat{p}(\mathbf{x})$. Finally, the above iterative optimization techniques can only guarantee achieving a local optimum,

so a branch and bound algorithm is necessary to achieve the global optimum at the cost of increasing computation burden (Liu, Shen and Wong, 2005).

4 Tuning via classification stability

The performance of the margin-based presence-only approaches may depend on the tuning parameter(s), such as C_1 and C_2 in (4) and (5), and hence that their classification performance needs to be optimized with appropriately selected tuning parameter(s). In the sequel, we denote $\hat{f}$ as $\hat{f}_\lambda$ to indicate its dependence on the tuning parameter(s) $\lambda = (C_1, C_2)$, and $\hat{\phi}_\lambda(\mathbf{x}) = \text{sign}(\hat{f}_\lambda(\mathbf{x}))$ as the corresponding classification decision function.

In classification, the performance of $\hat{\phi}_\lambda$ is often measured by its generalization error (GE),

$$GE(\hat{\phi}_\lambda) = E(I(Y \neq \hat{\phi}_\lambda(\mathbf{X}))), \tag{6}$$

where expectation is taken over the unknown joint distribution of $(\mathbf{X}, Y)$. In order to estimate the GE, the conventional CV procedure is commonly used, which uses a subset of data for training and the remaining for validation. However, its estimation accuracy can be severely deteriorated in presence-only data, since only presence data is available and the conventional CV may mistakenly prefer classification functions that are more inclined to predict presence. In fact, due to the lack of absence data, it is difficult, if not impossible, to accurately estimate $GE(\hat{\phi}_\lambda)$.

4.1 Tuning criterion

This subsection proposes a tuning criterion that assesses the classification accuracy of ϕ_λ through its classification stability. The idea of stability has been previously used in Meinshausen and Bühlman (2010) and Xin and Zhu (2012) as variable selection stability for

selecting the informative variables, and Wang (2010) as clustering stability for selecting the number of clusters. In this section, we extend the stability idea to classification stability, and develop a novel tuning criterion that is particularly suitable for presence-only data, since the classification stability does not require the class label of the absence data and thus can be estimated based on both presence and absence data for better tuning accuracy.

To slightly abuse notation, let ϕ_λ be a learning algorithm given λ, and $\hat{\phi}_\lambda$ be the estimated classification function learned by applying the algorithm ϕ_λ to a dataset.

Definition 1 *(Classification Stability) The stability of ϕ_λ is defined as*

$$\begin{aligned} s(\phi_\lambda; n) &= E(\mathrm{corr}(\hat{\phi}_\lambda(\mathbf{X}), \hat{\phi}^*_\lambda(\mathbf{X}))) \\ &= E\left(\frac{P(\hat{\phi}_\lambda(\mathbf{X}) = \hat{\phi}^*_\lambda(\mathbf{X}) = 1) - P(\hat{\phi}_\lambda(\mathbf{X}) = 1) P(\hat{\phi}^*_\lambda(\mathbf{X}) = 1)}{\mathrm{sd}(\hat{\phi}_\lambda(\mathbf{X}))\, \mathrm{sd}(\hat{\phi}^*_\lambda(\mathbf{X}))} \right), \end{aligned} \tag{7}$$

where the expectation is taken over all possible samples of size n, the probability is taken with respect to $\mathbf{X}$, and $\hat{\phi}_\lambda(\mathbf{x})$ and $\hat{\phi}^*_\lambda(\mathbf{x})$ are obtained by applying ϕ_λ to two independent samples of equal size n. We set $\mathrm{corr}(\hat{\phi}_\lambda(\mathbf{X}), \hat{\phi}^*_\lambda(\mathbf{X}))$ *to be 0 if* $\mathrm{sd}(\hat{\phi}_\lambda(\mathbf{X})) = 0$ *or* $\mathrm{sd}(\hat{\phi}^*_\lambda(\mathbf{X})) = 0$.

The key idea of classification stability is that if we repeatedly draw samples from the population and apply the given classification algorithm ϕ_λ, a good algorithm should produce estimated classification decision functions that do not vary much from one sample to another. Note that $\phi_\lambda(x)$ only takes value in $\{-1, 1\}$, so the agreement between $\hat{\phi}_\lambda(\mathbf{X})$ and $\hat{\phi}^*_\lambda(\mathbf{X})$ can be measured as $P(\hat{\phi}_\lambda(\mathbf{X}) = \hat{\phi}^*_\lambda(\mathbf{X}))$. However, this agreement measure can be misleading in assessing presence-only approaches as $P(\hat{\phi}_\lambda(\mathbf{X}) = \hat{\phi}^*_\lambda(\mathbf{X})) \equiv 1$ if ϕ_λ is degenerated and always assign observations to the presence class. Here we propose to use correlation in (7) as the agreement measure, since correlation standardizes the probability of classification agreement between $\hat{\phi}_\lambda(\mathbf{X})$ and $\hat{\phi}^*_\lambda(\mathbf{X})$ relative to their individual classifications, and is able to discriminate the truly stable classification algorithm from the seemingly stable algorithms that are stable only due to degenerated classification of all observations to the same class.

To estimate $s(\phi_\lambda; n)$ in practice, one can split the data into two training sets and one validation set. The two training sets are used to construct two estimated classification decision functions, and then their agreement on the left-out validation set estimates the classification stability. The splitting can be repeated multiple times, and the averaged classification stability then serves as the estimate of the classification accuracy of ϕ_λ. The proposed algorithm is described as follows.

Algorithm 1:

Step 1. Permute background data $(\mathbf{x}_{n_l+1}, \ldots, \mathbf{x}_n)$ and obtain $(\mathbf{x}^{*b}_{n_l+1}, \ldots, \mathbf{x}^{*b}_n)$.

Step 2. Split the permuted background data $(\mathbf{x}^{*b}_{n_l+1}, \ldots, \mathbf{x}^{*b}_n)$ into three parts with m, m and $n_u - 2m$ observations respectively: $\mathbf{x}^{*b}_I = (\mathbf{x}^{*b}_{n_l+1}, \ldots, \mathbf{x}^{*b}_{n_l+m})$, $\mathbf{x}^{*b}_{II} = (\mathbf{x}^{*b}_{n_l+m+1}, \ldots, \mathbf{x}^{*b}_{n_l+2m})$, and $\mathbf{x}^{*b}_{III} = (\mathbf{x}^{*b}_{n_l+2m+1}, \ldots, \mathbf{x}^{*b}_n)$.

Step 3. Train $\hat{\phi}^I_\lambda(\mathbf{x})$ and $\hat{\phi}^{II}_\lambda(\mathbf{x})$ based on $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n_l}$ with $\mathbf{x}^{*b}_I$ and $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n_l}$ with $\mathbf{x}^{*b}_{II}$, respectively. Letting $\mathbf{u}^b(\lambda) = (\hat{\phi}^I_\lambda(\mathbf{x}^{*b}_j))_{j=n_l+2m+1}^{n}$ and $\mathbf{v}^b(\lambda) = (\hat{\phi}^{II}_\lambda(\mathbf{x}^{*b}_j))_{j=n_l+2m+1}^{n}$, define $\hat{s}^{*b}(\hat{\phi}_\lambda; m)$ as the sample correlation between $\mathbf{u}^b(\lambda)$ and $\mathbf{v}^b(\lambda)$,

$$\hat{s}^{*b}(\phi_\lambda; m) = \frac{\sum_{j=1}^{n_u-2m}(u^b_j(\lambda) - \bar{u}^b(\lambda))(v^b_j(\lambda) - \bar{v}^b(\lambda))}{s^b_u(\lambda) s^b_v(\lambda)},$$

where $\bar{u}^b(\lambda)$, $\bar{v}^b(\lambda)$, $s^b_u(\lambda)$ and $s^b_v(\lambda)$ are the sample means and standard deviations of $\mathbf{u}^b(\lambda)$ and $\mathbf{v}^b(\lambda)$. If $s^b_u(\lambda) = 0$ or $s^b_v(\lambda) = 0$, $\hat{s}^{*b}(\phi_\lambda; m) = 0$.

Step 4. Repeat *Steps 1-3* for $b = 1, \ldots, B$, and define the estimated $s(\phi_\lambda; m)$ as

$$\hat{s}(\phi_\lambda; m) = B^{-1} \sum_{b=1}^{B} \hat{s}^{*b}(\phi_\lambda; m).$$

Then the tuning parameter λ can be selected as $\hat{\lambda}_m = \text{argmax}_\lambda \, \hat{s}(\phi_\lambda; m)$. In practice, one may implement a grid search scheme to approximate the global minimum $\hat{\lambda}_m$. Note that the proposed estimation scheme differs from the conventional CV in that it tries to estimate the

classification stability based on both presence and absence data as the classification stability does not require the class label of the absence data. Furthermore, the CV scheme can be replaced by other data re-sampling schemes, such as the boostrap as used in Meinshausen and Bühlman (2010).

4.2 Local consistency

We now establish the local consistency of the proposed tuning criterion, which assures that there exist a sequence of local maximizers of $\hat{s}(\phi_\lambda; m)$ such that they converge to the "optimal" λ with overwhelming probability in m.

Definition 2 *(Optimal λ) The optimal λ for ϕ is $\lambda_{m,o}$ if for any $\epsilon > 0$, there exists $a_\epsilon > 0$ such that for any λ_m with $\frac{\lambda_m}{\lambda_{m,o}} \to a \neq 1$ as $m \to \infty$, when m is sufficiently large,*

$$P\left(\frac{\text{corr}(\hat{\phi}_{\lambda_m}(\mathbf{X}), \hat{\phi}^*_{\lambda_m}(\mathbf{X}))}{\text{corr}(\hat{\phi}_{\lambda_{m,o}}(\mathbf{X}), \hat{\phi}^*_{\lambda_{m,o}}(\mathbf{X}))} \leq 1 - a_\epsilon\right) \geq 1 - \epsilon. \tag{8}$$

Definition 2 is analogous to Definition 3 in Wang (2010) for cluster analysis, which assures that the stability of $s(\phi_{\lambda_{m,o}}; m)$ is asymptotically greater than $s(\phi_{\lambda_m}; m)$ for any λ_m with $\frac{\lambda_m}{\lambda_{m,o}} \to a \neq 1$. The relative larger magnitude of $s(\phi_{\lambda_{m,o}}; m)$ in (8) does not necessarily require that it converges to 1 at a faster rate than other candidates. In fact, even when $s(\phi_{\lambda_{m,o}}; m)$ converges to 1 at the same rate as other candidates, a larger constant in its rate of convergence is sufficient for (8).

We outline the main theorem and assumption here, and defer the technical details to the Appendix.

Assumption 1. Assume that $s(\phi_\lambda; m)$ converges to 1 exactly at rate $r_{m,\lambda}$ in probability, where $r_{m,\lambda}$ is a sequence of non-increasing positive numbers.

The "exact convergence" is defined in the sense of Definition 2 of Yang (2006), which guarantees that $s(\phi_\lambda; m)$ does not converge to 1 faster than the given rate on a set with

positive probability. In the literature, the rates of convergence of many SSL approaches have been established, such as Rigollet (2007) and Wang et al. (2009).

Theorem 1 *Suppose $\lambda_{m,o}$ exists and Assumption 1 holds. There exist a sequence of local maximizers of $\hat{s}(\phi_\lambda; m)$, $\hat{\lambda}_m$, such that*

$$\frac{\hat{\lambda}_m}{\lambda_{m,o}} \to 1 \text{ in probability,}$$

as long as $m \to \infty$ and $(n-2m)\min_{\frac{\lambda_m}{\lambda_{m,o}} \to a \neq 1} r^2_{m,\lambda_m} \to \infty$.

Theorem 1 establishes the local consistency of the proposed tuning criterion when the data is properly split. It also provides guideline for splitting the data in Algorithm 1 in order to achieve the local consistency. In specific, if B is fixed as a constant, and $s(\phi_\lambda; m)$ converges to 1 exactly at rate $O_p(m^{-1/2})$ for all λ_m, then the requirement on the data splitting ratio becomes $(n-2m)/m \to \infty$, which agrees with Shao (1993) for linear regression and Yang (2006) for classification. If $s(\phi_\lambda; m)$ converges to 1 at a faster rate than $O_p(m^{-1/2})$ for some λ_m such as in Wang et al. (2009), $(n-2m)/m$ needs to diverge at a faster rate as well, while if $s(\phi_\lambda; m)$ converges to 1 at a slower rate than $O_p(m^{-1/2})$ for all λ_m, $(n-2m)/m = O(1)$ will suffice.

5 Numerical experiment

This section examines the numerical performance of the proposed PO-TSVM and the PO-ESSL as well as other existing presence-only approaches, including the naive SVM (Naive), the iterative SVM (Iter) and the EM algorithm (EM; Ward et al., 2009). Note that the PO-ESSL can be initialized by any presence-only approach, and we denote the PO-ESSL approach initialized by naive SVM, iterative SVM and the PO-TSVM, as PO-ENaive, PO-EIter and PO-ETSVM respectively.

5.1 Simulation examples

Three simulated presence-only examples are considered: two-Gaussian, two-moon and bull's eye examples. The data distributions of all examples are displayed in Figure 2. The two-Gaussian example has 10 dimensions, where only the first two dimensions are displayed and the remaining 8 dimensions are noise variables generated from standard normal distribution. The other two examples have 2 dimensions as it is often much more difficult to learn nonlinear classification functions based on only a few presence data.

Insert Figure 2 about here

For each simulated example, 1000 sample points are randomly generated and divided into halves, with 10 presence and 90 background data points for training and the remaining 900 data points for testing. A test error measured on the test set,

$$TE(\hat{\phi}) = \frac{1}{\#\{\text{test set}\}} \sum_{\text{test set}} I(y_i \neq \hat{\phi}(\mathbf{x}_i)),$$

is used to measure the classification performance of all methods in comparison, where $\#\{\text{test set}\}$ is the cardinality of the test set.

As Figure 2 suggests, the ideal classification function in the two-Gaussian example is linear, whereas that in the two-moon and the bull's eye examples are nonlinear. Therefore, we construct the linear SVM for the two-Gaussian example, and the SVM with Gaussian kernel for the two-moon and bull's eye examples. When Gaussian kernel is used, the standard deviation is set to be the median pairwise Euclidean distance among all training data to reduce computational cost for tuning, c.f., Jaakkola et al. (1999). Furthermore, the EM algorithm is implemented following Ward et al. (2009), which requires prior knowledge about the proportion of the presence sample that is often unavailable in practice. In the simulation examples, we set it to be the true proportion 1/2. The linear logistic regression

and the kernel logistic regression (Zhu and Hastie, 2005) are used in the EM algorithm for the two-Gaussian example and the other two nonlinear examples respectively.

To eliminate the dependence of the classifier on other tuning parameters, three tuning criteria are examined. The first one is the estimated GE pretending that the labels of all the background data are available, the second one is the estimated classification stability as described in *Algorithm 1*, and the last one is 5-fold CV. Although the first criterion is unrealistic in practice, it is compared to the other two criteria to examine their effectiveness in tuning presence-only approaches. A grid search scheme is performed to optimize each tuning criterion. Specifically, one tuning parameter for the naive SVM and the iterative SVM, two tuning parameters for the PO-TSVM and the PO-ESSL are searched over grid points $10^{-2+k/3}; k = 0, \ldots, 12$. Finally, the averaged test errors over 100 independent replications are summarized in Table 1.

Insert Table 1 about here

Evidently, the PO-ESSL approaches, including PO-ENaive, PO-EIter and PO-ETSVM, yield superior performance over their counterparts in all examples, and they improve the classification performance of their initializers respectively. In particular, PO-ETSVM appears to be the most competitive performer as it yields the smallest or the second smallest test errors in most scenarios. The EM algorithm appears to work well in the two-Gaussian examples, but less satisfactory in the other two nonlinear examples.

In addition, the proposed tuning criterion via classification stability outperforms 5-fold CV in almost all scenarios, and yields comparable test errors to those with tuning parameters selected via the labels of the background data. Note that the EM algorithm does not require tuning, so it yields the same test errors no matter what tuning criterion is employed. To scrutinize the relationship between these two tuning criteria, we examine one randomly chosen replication in Example 1 for PO-ETSVM. As displayed in Figure 3, it is clear that

large value of estimated classification stability imply low estimated GE based on the labels of background data, which confirms the satisfactory performance of estimated classification stability in Table 1, and demonstrates the effectiveness of classification stability in selecting tuning parameters for presence-only data. In Figure 3, we also plot the estimated GE via 5-fold CV, which is clearly not a desirable criterion for tuning PO-ETSVM due to the lack of negative observations.

Insert Figure 3 about here

5.2 Gene function prediction

This section applies the proposed presence-only SSL approach to predict gene functions based on the gene data in Hughes et al. (2000), consisting of expression profiles of a total of 6316 genes for yeast S. *cerevisiae* from 300 microarray experiments. The gene functional categories are defined by the MIPS, a multifunctional classification scheme Mewes et al. (2002). The microarray gene expression profiles can be used to predict gene functions, because genes sharing the same function tend to co-express, c.f., Zhou, Kao and Wong (2000). Unfortunately, based on the available biological information, we know which genes are annotated by the function of interest, but it is generally unclear which genes do not have the function. Therefore, it is appropriate to predict the gene function class of unannotated gene through presence-only approaches.

Note that assessing classification performance of presence-only approaches in real application is difficult as no validation dataset with true presence and absence is available. To alleviate this difficulty, we generate a presence-only data from the available dataset in Hughes et al. (2000), and focus on two popular functional categories, namely "TRANSCRIPTION" and "PROTEIN FATE". These two gene functions annotate 578 and 533 genes respectively, whose gene expression profiles based on 300 microarray experiments are also available. To mimic the presence-only scenario, a small portion of genes annotated by "PROTEIN FATE"

are treated as presence data, and all other genes are treated as background data with unknown functions. In particular, we divide the 1111 genes into a training set and a test set, where the training set involves a random sample of n_l presence data and $400-n_l$ background data, while the remaining 711 genes are used for testing. To examine the sensitivity of the presence-only approaches to the size of presence sample, $n_l = 20, 50, 100, 150$ and 200 are tried. The splitting is repeated 100 times, and the tuning parameters are estimated through the same grid search scheme as in Section 5.1. The averaged test errors are summarized in Table 2. Additionally, as a baseline for comparing classification performance, the test error of a full model is reported in Table 2, which fits a SVM model with Gaussian kernel using the complete labels of the background data.

Insert Table 2 about here

As showed in Table 2, PO-ENaive, PO-EIter and PO-ETSVM outperform their initial counterparts respectively. The classification accuracies of PO-ENaive and PO-ETSVM improve as n_l increases, and when n_l is reasonably large they yield comparable classification performance to that of the full model using the complete labels of the background data. However, the performance of Iter gets worse as n_l increases, which deteriorates the performance of PO-EIter as well. Note that the EM algorithm based on logistic regression (Ward et al., 2009) is not applied in this gene example as the logistic regression fails to converge when fitting a high dimensional dataset ($d = 300$) with relatively small sample size ($n = 400$).

6 Summary

This article proposes to analyzes the presence-only data through SSL approaches. To overcome the difficulty of unavailable absence data, a class balance constraint is enforced to guard the estimated classification function from degeneration, and a novel model tuning criterion

based on classification stability is proposed for optimizing the predictive performance of classification. The numerical results on a variety of simulation examples and a real example on gene function prediction suggest that the proposed method delivers desirable classification performance and compares favorably against top competitors.

Appendix: technical proofs

Proof of Theorem 1: For any $\eta > 0$, we first focus on the b-th replication, and compare $\hat{s}^{*b}(\phi_{\lambda_{m,o}}; m)$ to $\hat{s}^{*b}(\phi_{\lambda_k}; m)$; $k = 1, 2$ with $\lambda_1 = \lambda_{m,o} - \eta$ and $\lambda_2 = \lambda_{m,o} + \eta$. By Assumption 1, we have that for any arbitrary $\epsilon > 0$, there exists $a_\epsilon > 0$ such that $P(A_k^c) \leq \epsilon$ with $A_k = \{\text{corr}(\hat{\phi}_{\lambda_k}(\mathbf{X}), \hat{\phi}^*_{\lambda_k}(\mathbf{X})) \leq (1 - a_\epsilon)\,\text{corr}(\hat{\phi}_{\lambda_{m,o}}(\mathbf{X}), \hat{\phi}^*_{\lambda_{m,o}}(\mathbf{X}))\}$; $k = 1, 2$. On A_k, conditional on $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n_l}$, $\mathbf{x}_I^{*b}$ and $\mathbf{x}_{II}^{*b}$,

$$
\begin{aligned}
& P\Big(\hat{s}^{*b}(\phi_{\lambda_{m,o}}; m) < \hat{s}^{*b}(\phi_{\lambda_k}; m)\Big|\{(\mathbf{x}_i, y_i)\}_{i=1}^{n_l}, \mathbf{x}_I^{*b}, \mathbf{x}_{II}^{*b}\Big) \\
= \; & P\left(\sum_{j=1}^{n_u-2m} W_{kj} > 0\Big|\{(\mathbf{x}_i, y_i)\}_{i=1}^{n_l}, \mathbf{x}_I^{*b}, \mathbf{x}_{II}^{*b}\right) \\
= \; & P\left(\sum_{j=1}^{n_u-2m} (W_{kj} + \Delta_k) > (n_u - 2m)\Delta_k\Big|\{(\mathbf{x}_i, y_i)\}_{i=1}^{n_l}, \mathbf{x}_I^{*b}, \mathbf{x}_{II}^{*b}\right),
\end{aligned}
$$

where $W_{kj} = \frac{(u_j(\lambda_k) - \bar{u}(\lambda_k))(v_j(\lambda_k) - \bar{v}(\lambda_k))}{s_u(\lambda_k)s_v(\lambda_k)} - \frac{(u_j(\lambda_{m,o}) - \bar{u}(\lambda_{m,o}))(v_j(\lambda_{m,o}) - \bar{v}(\lambda_{m,o}))}{s_u(\lambda_{m,o})s_v(\lambda_{m,o})}$ as in Algorithm 1, and

$$
\begin{aligned}
\Delta_k &= -E(W_{kj}\Big|\{(\mathbf{x}_i, y_i)\}_{i=1}^{n_l}, \mathbf{x}_I^{*b}, \mathbf{x}_{II}^{*b}) \\
&= \text{corr}(\hat{\phi}_{\lambda_{m,o}}(\mathbf{X}), \hat{\phi}^*_{\lambda_{m,o}}(\mathbf{X})) - \text{corr}(\hat{\phi}_{\lambda_k}(\mathbf{X}), \hat{\phi}^*_{\lambda_k}(\mathbf{X})) + O_p((n_u - 2m)^{-1}) \\
&\geq a_\epsilon\,\text{corr}(\hat{\phi}_{\lambda_{m,o}}(\mathbf{X}), \hat{\phi}^*_{\lambda_{m,o}}(\mathbf{X})) + O_p((n_u - 2m)^{-1}).
\end{aligned}
$$

Applying the Bernstein's inequality (Pollard, 1984) yields that on A_k,

$$P\Big(\hat{s}^{*b}(\phi_{\lambda_{m,o}};m) < \hat{s}^{*b}(\phi_{\lambda_k};m)\Big|\{(\mathbf{x}_i,y_i)\}_{i=1}^{n_l},\mathbf{x}_I^{*b},\mathbf{x}_{II}^{*b}\Big) \leq \exp\left(-\frac{(n_u-2m)\Delta_k^2}{2V_k+\frac{4}{3}\Delta_k}\right),$$

where $V_k = \mathrm{var}(w_{kl}|\{(\mathbf{x}_i,y_i)\}_{i=1}^{n_l},\mathbf{x}_I^{*b},\mathbf{x}_{II}^{*b}) \leq 4^2$ and $\Delta_k < 3$. Therefore, for any given k,

$$P(\hat{s}^{*b}(\phi_{\lambda_{m,o}};m) < \hat{s}^{*b}(\phi_{\lambda_k};m)) \leq \epsilon + \exp\left(-\frac{(n_u-2m)\Delta_k^2}{36}\right),$$

which implies that

$$\begin{aligned}
&\sum_{k=1}^{2} P\left(\sum_{b=1}^{B}\hat{s}^{*b}(\phi_{\lambda_{m,o}};m) < \sum_{b=1}^{B}\hat{s}^{*b}(\phi_{\lambda_k};m)\right) \\
\leq\ &\sum_{k=1}^{2}\sum_{b=1}^{B} P\Big(\hat{s}^{*b}(\phi_{\lambda_{m,o}};m) < \hat{s}^{*b}(\phi_{\lambda_k};m)\Big) \leq 2B\epsilon + \sum_{k\neq k_o} B\exp\left(-\frac{(n_u-2m)\Delta_k^2}{36}\right).
\end{aligned}$$

Let $\epsilon = 1/mB$, then $2B\epsilon$ converges to 0 as $m\to\infty$. Furthermore, Assumption 1 implies that Δ_k converges to 0 exactly at rate r_{m,λ_k}, and hence that $(n_u-2m)\Delta_k^2 \to \infty$ if $(n-2m)r_{m,\lambda_k}^2 \to \infty$. Therefore, when $m\to\infty$ and $(n-2m)r_{m,\lambda_k}^2\to\infty$,

$$P(\hat{s}^{*b}(\phi_{\lambda_{m,o}};m) < \hat{s}^{*b}(\phi_{\lambda_k};m)) \to 0;\ \text{for } k=1,2. \tag{9}$$

This implies that with probability approaching 1, there exists a local maximum of $\hat{s}^{*b}(\phi_\lambda;m)$ in $(\lambda_{m,o}-\eta,\lambda_{m,o}+\eta)$ for any $\eta>0$. The desired results follows by setting $\eta = 1/m$.

References

[1] Belkin, M. and Niyogi, P. (2004). Semi-supervised Learning on Riemannian Manifolds. *Mach. Learn.*, **56**, 209-239.

[2] Belkin, M., Niyogi, P. and Sindhwani, V. (2006).Manifold Regularization: a Geometric Framework for Learning from Labeled and Unlabeled Examples. *J. Mach. Learn. Res.*, **7**, 2399-2434.

[3] Chapelle, O., Sindhwani, V. and Keerthi, S. (2008). Optimization Techniques for Semi-Supervised Support Vector Machines. *J. Mach. Learn. Res.*, **9**, 203-233.

[4] Chapelle, O. and Zien, A. (2005). Semi-supervised classification by low density separation. In *Proc. Int. Workshop on Artif. Intel. and Statist.*, 57-64.

[5] Cortes, C. and Vapnik, V. (1995). Support vector networks. *Mach. Learn.*, **20**, 273-297.

[6] Guan, N., Tao, D., Luo, Z. and Yuan, B. NeNMF: An Optimal Gradient Method for Nonnegative Matrix Factorization. *IEEE Trans. Sig. Processing*, **60**, 2882-2898.

[7] Hughes, T., Marton, M., Jones, A., Roberts, C., Stoughton, R., Armour, C., Bennett, H., Coffey, E., Dai, H., He, Y., Kidd, M., King, A., Meyer, M., Slade, D., Lum, P., Stepaniants, S., Shoemaker, D., Gachotte, D., Chakraburtty, K., Simon, J., Bard, M. and Friend, S. (2000). Functional discovery via a compendium of expression profiles. *Cell*, **102**, 109-126.

[8] Jaakkola, T., Diekhans, M. and Haussler, D. (1999). Using the Fisher kernel method to detect remote protein homologies. In *Proc. Int. Conf. on Intelligent Systems for Molecular Biology*, 149-158.

[9] Joachims, T. Transductive Inference for Text Classification using Support Vector Machines. In *Proc. 16th Int. Conf. Machine Learning (ICML)*, pp. 200-209. Morgan Kaufmann, San Francisco.

[10] Keating, K. and Cherry, S. (2004). Use and interpretation of logistic regression in habitat-selection studies. *J. Wildl. Manage.*, **68**, 774-789.

[11] Liu, B., Dai, Y., Li, X., Lee, W. and Yu, P. (2003). Building Text Classifiers Using Positive and Unlabeled Examples. *Inter. Conf. on Data Mining (ICDM).*

[12] Liu, S., Shen, X. and Wong, W. (2005). Computational development of ψ-learning. In *Proc. SIAM 2005 Inter. Data Mining Conf.*, 1-12.

[13] Meinshausen, N. and Buhlman, P. Stability selection (with discussion). *J. Royal Statist. Soc. Ser. B*, **72**, 417-473.

[14] Mewes, H., Albermann, K., Heumann, K., Liebl S. and Pfeiffer, F. (2002). MIPS: a database for protein sequences, homology data and yeast genome information. *Nucleic Acids Res.*, **25**, 28-30.

[15] Pollard, D. (1984). Convergence of Stochastic Processes. Springer

[16] Rigollet, P. (2007). Generalization Error Bounds in Semi-supervised Classification Under the Cluster Assumption. *J. Mach. Learn. Res.*, **8**, 1369-1392.

[17] Shao, J. (1993). Linear model selection by cross-validation. *J. Amer. Statist. Assoc.*, **88**, 486-494.

[18] Tian, X., Tao, D. and Rui, Y. (2012). Sparse Transfer Learning for Interactive Video Search Reranking. *ACM Trans. Multimedia Computing, Communications and Applications*, to appear.

[19] Vapnik, V. (1998). *Statistical Learning Theory*, Wiley, New York.

[20] Wang, J. (2010). Consistent selection of the number of clusters via cross-validation. *Biometrika*, **97**, 893-904.

[21] Wang, J., Shen, X. and Liu, Y. (2008). Probability estimation for large margin classifiers. *Biometrika*, **95**, 149-167.

[22] Wang, J., Shen, X. and Pan, W. (2007). On transductive support vector machine. *Contemp. Math.*, **43**, 7-19.

[23] Wang, J., Shen, X. and Pan, W. (2009). On efficient large margin semisupervised learning: methodology and theory. *J. Mach. Learn. Res.*, **10**, 719-742.

[24] Ward, G., Hastie, T., Barry, S., Elith, J. and Leathwick, J. (2009). Presence-Only Data and the EM Algorithm. *Biometrics*, **65**, 554-563.

[25] Xin, L. and Zhu, M. (2012). Stochastic stepwise ensembles for variable selection. *J. Comput. Graph. Statist.*, **21**, 275-294.

[26] Yang, Y. (2006). Comparing learning methods for classification. *Statist. Sinica*, **16**, 635-657.

[27] Zhao, X., Wang, Y., Chen, L. and Aihara, K. (2008). Gene function prediction using labeled and unlabeled data. *BMC Bioinformatics*, **9**, 1471C2105.

[28] Zhou, X., Kao, M. and Wong, W. (2000). Transitive functional annotation by shortest-pathanalysis of gene expression data. *Proc. Nat. Acad. Sci.*, **99**, 12783-12788.

[29] Zhu, J. and Hastie, T. (2005). Kernel logistic regression and the import vector machine. *J. Comput. Graph. Statist.*, **14**, 185-205.

[30] Zhu, X. Semi-supervised learning literature survey. Technical Report 1530, University of Wisconsin, Madison, 2005.

[31] Zhu, X., Ghahramani, Z. and Lafferty, J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. *Int. Conf. on Mach. Learn. (ICML)*.

Table 1: Simulated examples: the averaged test errors and the estimated standard errors in parenthesis. The smallest test errors in each scenario are boldfaced.

Examples	Naive	Iter	PO-TSVM	PO-ENaive	PO-EIter	PO-ETSVM	EM
Tuned with labels of background data							
Two-Gaussian	.460 (.0032)	.113 (.0067)	.127 (.0095)	.171 (.0064)	**.095** (.0055)	.107 (.0092)	.109 (.0048)
Two-Moon	.415 (.0042)	.107 (.0071)	.082 (.0028)	.035 (.0039)	.035 (.0049)	**.020** (.0033)	.098 (.0043)
Bull's Eye	.442 (.0045)	.152 (.0066)	.126 (.0039)	.114 (.0056)	.098 (.0061)	**.070** (.0049)	.125 (.0073)
Tuned with classification stability (Algorithm 1)							
Two-Gaussian	.460 (.0032)	.180 (.0128)	.166 (.0107)	.194 (.0076)	**.110** (.0068)	.123 (.0100)	.132 (.0083)
Two-Moon	.445 (.0041)	.140 (.0058)	.131 (.0052)	**.056** (.0091)	.075 (.0087)	.069 (.0079)	.112 (.0091)
Bull's Eye	.439 (.0046)	.229 (.0086)	.134 (.0107)	.169 (.0068)	.168 (.0082)	**.096** (.0091)	.186 (.0110)
Tuned with 5-fold cross validation							
Two-Gaussian	.460 (.0032)	.194 (.0122)	.199 (.0108)	.184 (.0065)	.150 (.0091)	.146 (.0118)	**.141** (.0098)
Two-Moon	.445 (.0041)	.248 (.0144)	.129 (.0097)	.135 (.0083)	.140 (.0109)	**.120** (.0076)	.137 (.0105)
Bull's Eye	.439 (.0046)	.222 (.0116)	.170 (.0144)	.170 (.0060)	.144 (.0103)	**.115** (.0091)	.210 (.0138)

Table 2: Gene function prediction: the averaged test errors and the estimated standard errors in parenthesis. The smallest test errors in each scenario are boldfaced.

Gene	Naive	Iter	PO-TSVM	PO-ENaive	PO-EIter	PO-ETSVM	Full
$n_l = 20$	.483 (.0013)	.460 (.0080)	.426 (.0047)	.484 (.0013)	.451 (.0034)	**.415** (.0106)	
$n_l = 50$	.483 (.0011)	.456 (.00104)	.419 (.0038)	.471 (.0014)	.420 (.0037)	**.392** (.0054)	
$n_l = 100$	.481 (.0011)	.444 (.0051)	.406 (.0036)	.443 (.0033)	.396 (.0059)	**.377** (.0039)	.301 (.0014)
$n_l = 150$	.360 (.0012)	.475 (.0037)	.396 (.0067)	**.332** (.0016)	.446 (.0024)	.379 (.0050)	
$n_l = 200$	.317 (.0018)	.479 (.0019)	.336 (.0016)	**.322** (.0038)	.481 (.0015)	.338 (.0030)	

Figure 1: The testing errors of TSVM with equality or inequality constraints, as functions of the tuning parameter D. Here other tuning parameters are fixed as $C_1 = 100$ and $C_2 = 1$ for illustration.

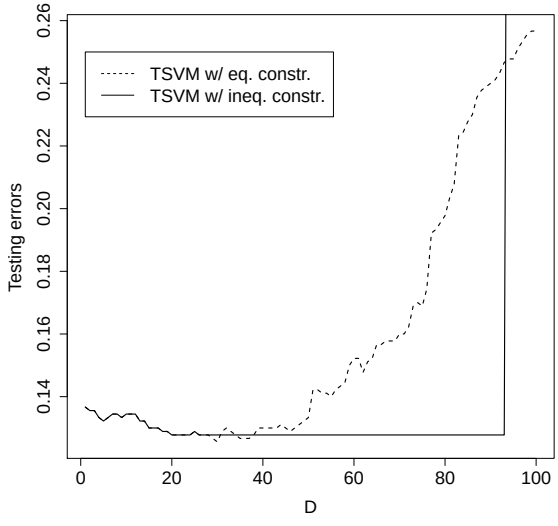

Figure 2: The plots of the two-Gaussian, two-moon and bull's eye data.

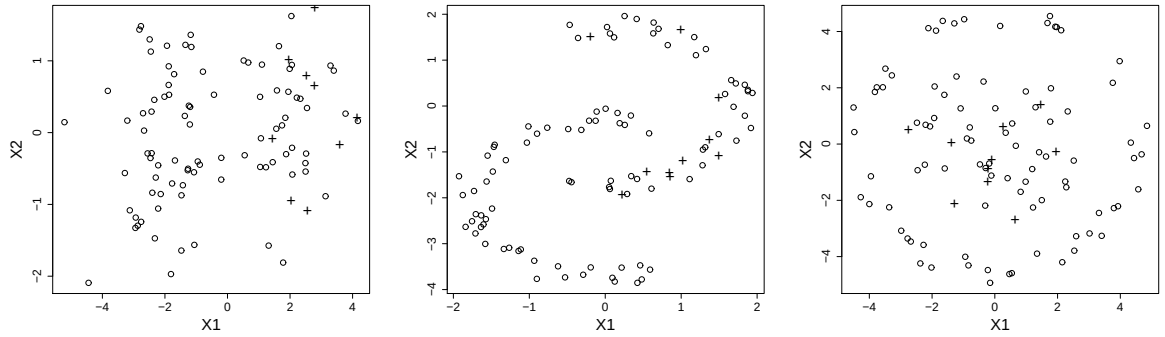

Figure 3: Plots of the estimated classification stability of ETSVM, the estimated GE via labels of background data, and the estimated GE via 5-fold cross validation as functions of tuning parameters. In the left panel, C_2 is fixed as 1; and in the right panel, C_1 is fixed as 10.