Using adaptively weighted large margin classifiers for robust sufficient dimension reduction

Andreas Artemiou
School of Mathematics, Cardiff University

Abstract

In this paper we combine adaptively weighted large margin classifiers with Support Vector Machine (SVM)-based dimension reduction methods to create dimension reduction methods robust to the presence of extreme outliers. We discuss estimation and asymptotic properties of the algorithm. The good performance of the new algorithm is demonstrated through simulations and real data analysis.

Key Words: Dimension reduction; adaptive weights; Support Vector Machines; Outliers

1 Introduction

Nowadays, high dimensional problems are becoming the norm due to the increase of computing power and storage capabilities. At the same time classic statistical techniques, which were developed based on low dimensional problems, lack the ability to generalize and perform robustly in high dimensional problems. One way to overcome this difficulty is to perform dimension reduction to our data before applying any of the traditional techniques to it.

Sufficient Dimension Reduction (SDR) is a class of techniques for supervised feature extraction in a high dimensional regression (or classification) setting. In SDR we assume that we have a univariate (without loss of generality) response variable Y and a p dimensional predictor vector $\boldsymbol{X}$. Our objective is to estimate a set of d features (where $d \leq p$) without losing information on the conditional distribution of $Y|\boldsymbol{X}$. In other words, we are trying to estimate a $p \times d$ matrix $\boldsymbol{\beta}$ which satisfies

$$Y \perp\!\!\!\perp \boldsymbol{X} | \boldsymbol{\beta}^{\mathsf{T}} \boldsymbol{X}. \tag{1}$$

Since the extracted features are linear functions of the original predictors this is called linear SDR. The space spanned by the columns of $\boldsymbol{\beta}$ is called the Dimension Reduction Subspace (DRS). The intersection of all possible DRSs if it is itself a DRS it is called the Central Dimension Reduction Subspace (CDRS) or simply the Central Subspace (CS) and it is denoted with $\mathcal{S}_{Y|\boldsymbol{X}}$. CS is the space that has the smaller dimension (d) among all DRSs. Although the CS doesn't always exist the assumptions required for existence are mild so for the rest of the paper we assume existence of the CS (see Cook - 1998a). Classic methods in the SDR literature have been proposed in Li (1991), Cook and Weisberg (1991), Li (1992), Cook (1998b), Li, Zha, Chiaromonte (2005) and Li and Wang (2007) among others.

More recently, Li, Artemiou and Li (2011) have proposed Principal Support Vector Machine (PSVM) which uses previous ideas in the SDR framework as well as Support Vector Machine (SVM) to achieve dimension reduction. The most important advantage of this algorithms is that it provides a common framework for linear and nonlinear SDR. Artemiou and Shu (2014) applied a cost based reweight technique which improved the performance of the algorithm as it was taking into account the imbalanced nature between the slices. Moreover, Shin et al (2014, 2017) have used weights to estimate a probability enhanced CDRS when the response is binary.

In this work, we are interested to develop a method that is robust to the presence of extreme outliers. Towards this, we introduce adaptive weights in the objective function of PSVM. To achieve this, we use the idea by Wu and Liu (2013) where adaptive weights are used to improve the classification performance of SVM (the most well-known large margin classifier). Wu and Liu (2013) proposed a two-run method. In the first run, they solve the optimization problem to obtain a first estimate of the optimal separating hyperplane. Then they suggested using a second run to find the final estimate of the optimal separating hyperplane. In the second run though they suggested to use the misclassification distance of the misclassified points as an inverse weight in the optimization problem. In the SDR framework, we propose to take a similar approach with Wu and Liu (2013) to improve the estimation performance of PSVM for dimension reduction. The new algorithm is called Adaptively Weighted Principal Support Vector Machine (AWPSVM). Further to this, we also apply the adaptive weights to Principal L2 SVM (PL2SVM) which was proposed by Artemiou and Dong (2016) and it was demonstrated that it generally has better performance than PSVM. Finally, although the theoretical framework of our methodology is similar to the one by Shin et al (2017) who used weighted SVM to achieve dimension reduction on binary responses, we emphasize that there are important differences. First of all, we have a different objective as we are targeting extreme outliers while they target dimension reduction when the response is binary. Furthermore, our methodology slices the response, which is not the case for Shin et al (2017).

The rest of the paper is organized as follows. In section 2 we discuss PSVM and other similar existing methodology and we introduce AWPSVM in section 3. In section 4 we discuss some asymptotic properties. In section 5 we present some simulation results and real data analysis follows in section 6. A small discussion closes the paper.

2 Previous work

In this section we discuss briefly different methods that were introduced in the SVM-based dimension reduction literature and which are related to the method we are proposing in this work. For the rest of the section suppose $(\boldsymbol{X}_i, Y_i)$ $i = 1, \dots, n$ independent observations. Let $\boldsymbol{\Sigma} = \mathrm{var}(\boldsymbol{X})$ and assume that the support of Y can be split in two disjoint sets A_1 and A_2 so that we define $\tilde{Y} = I(Y \in A_2) - I(Y \in A_1)$.

2.1 Principal Support Vector Machine (PSVM)

PSVM (Li, Artemiou and Li -2011) minimizes the following objective function:

$$L(\boldsymbol{\psi}, t) = \boldsymbol{\psi}^{\mathsf{T}}\boldsymbol{\Sigma}\boldsymbol{\psi} + \lambda E\{1 - \tilde{Y}[\boldsymbol{\psi}^{\mathsf{T}}(\boldsymbol{X} - E\boldsymbol{X}) - t]\}^{+}, \tag{2}$$

where λ is the misclassification penalty (or cost as it is known in the machine learning literature) and $a^{+} = \max\{0, a\}$. Also $(\boldsymbol{\psi}, t) \in \mathbb{R}^p \times \mathbb{R}$ define the equation of the separating hyperplane. The objective is to find a pair of $(\boldsymbol{\psi}^*, t^*) \in \mathbb{R}^p \times \mathbb{R}$ which minimizes the objective function (2). At the sample level the authors (roughly) suggested the use of different cutoff points q_k, $k = 1, \dots, h$ to construct multiple hyperplanes described by $(\hat{\boldsymbol{\psi}}_i^*, \hat{t}_i^*)$, $i = 1, \dots, h$. Then an eigenvalue decomposition of the matrix $\hat{\boldsymbol{M}} = \sum_{i=1}^{h} \hat{\boldsymbol{\psi}}_i^* (\hat{\boldsymbol{\psi}}_i^*)^{\mathsf{T}}$ will give us the eigenvectors corresponding to the largest d eigenvalues, where d is the estimated dimension of the CS.

2.2 Principal Lq Support Vector Machine

PSVM algorithm in Li, Artemiou and Li (2011) gave a unique solution of the optimal hyperplane in terms of the normal vector $\boldsymbol{\psi}$. This, though, was not true for the offset t. Therefore, Artemiou and Dong (2016) proposed the use of Lq Principal Support Vector Machine (LqSVM) in sufficient dimension reduction. The objective function in this case is:

$$L_2(\boldsymbol{\psi}, t) = \boldsymbol{\psi}^\intercal \boldsymbol{\Sigma} \boldsymbol{\psi} + E\{(1 - \tilde{Y}[\boldsymbol{\psi}^\intercal(\boldsymbol{X} - E\boldsymbol{X}) - t])^+\}^2, \tag{3}$$

where we have a strictly convex function that can ensure the uniqueness of the optimal hyperplane in both $\boldsymbol{\psi}$ and t. Although t is not used in the estimation of the CS, as one can see in both Li, Artemiou and Li (2011) and Artemiou and Dong (2016) it is important on the development of the asymptotic theory as different quantities (i.e. Hessian matrix and therefore asymptotic variance) depend on it's value.

2.3 Principal Weighted Support Vector Machine

Shin et al (2017) presented the following idea to achieve sufficient dimension reduction in cases with binary response:

$$L_W(\boldsymbol{\psi}, t) = \boldsymbol{\psi}^\intercal \boldsymbol{\Sigma} \boldsymbol{\psi} + \lambda E\{\pi(Y)(1 - Y[\boldsymbol{\psi}^\intercal(\boldsymbol{X} - E\boldsymbol{X}) - t])\}^+, \tag{4}$$

where $\pi(Y) = 1 - \pi$ and $\pi \in (0, 1)$. They incorporated weights using the idea of weighted SVM (see Lin et al (2002)) to estimate the CS. Here we emphasize that their method mainly tackles cases where there is binary response. Classic SDR methods cannot estimate more than one direction whenever the response is binary. On the other hand, the use of π-path trajectories in the weighted SVM algorithm helps avoid this issue in Shin et al (2017).

2.4 Cost Reweighted Principal Support Vector Machine

Artemiou and Shu (2014) presented another form of weighted algorithm. Their objective was to accommodate for cases where there was imbalance in the number of observations between the two disjoint sets A_1 and A_2 of the support of Y. The objective function in this case is:

$$L_{CR}(\boldsymbol{\psi}, t) = \boldsymbol{\psi}^\intercal \boldsymbol{\Sigma} \boldsymbol{\psi} + E\{\lambda_{\tilde{Y}}(1 - \tilde{Y}[\boldsymbol{\psi}^\intercal(\boldsymbol{X} - E\boldsymbol{X}) - t])\}^+, \tag{5}$$

where the only difference from the PSVM method is the dependence of the misclassification penalty λ on the value of $\tilde{Y}$ to show that the two classes have different costs. Again the objective of this algorithm was to use cost based reweighting to target bias introduced due to imbalance and not to address the presence of extreme outliers as we do in this case.

3 Adaptively weighted algorithms for SDR

In this section we propose adaptively weighted versions of PSVM from Li, Artemiou and Li (2011) and Principal L2SVM from Artemiou and Dong (2016). To achieve this we use the adaptively weighted SVM idea in Wu and Liu (2013).

3.1 Adaptively Weighted Principal SVM

Adaptively Weighted Principal Support Vector Machine introduce weights into the objective function. This weights are carefully chosen so that extreme outliers, and more specifically points that are incorrectly classified and further away from the separating hyperplane, get a smaller weight and their importance is downplayed. The objective function takes the form

$$L_{AW}(\boldsymbol{\psi}, t) = \boldsymbol{\psi}^{\mathsf{T}}\boldsymbol{\Sigma}\boldsymbol{\psi} + \lambda E\{w(1 - \tilde{Y}[\boldsymbol{\psi}^{\mathsf{T}}(\boldsymbol{X} - E\boldsymbol{X}) - t])^{+}\}, \tag{6}$$

where for this work we assume that $w > 0$ (we will discuss the choice of the weights in the estimation section). Also notice how there is a similarity of this with both the weighted algorithms discussed in the previous section.

In the next theorem we show that indeed one can use the $\boldsymbol{\psi}^{*} \in \mathbb{R}^{p}$ to estimate the CDRS. The proof is similar to the respective theorems in Li, Artemiou, Li (2011), Artemiou and Shu (2014) and Shin et al (2017).

Theorem 1 *Suppose $E(\boldsymbol{X}|\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{X})$ is a linear function of $\boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{X}$, where $\boldsymbol{\beta}$ is as defined in (1). If $(\boldsymbol{\psi}^{*}, t^{*})$ minimizes the objective function (6) among all $(\boldsymbol{\psi}, t) \in \mathbb{R}^{p} \times \mathbb{R}$, then $\boldsymbol{\psi}^{*} \in \mathcal{S}_{Y|\boldsymbol{X}}$.*

Proof. From the population version in (6) let's assume without loss of generality that $E(\boldsymbol{X}) = 0$ so it becomes

$$L_{AW}(\boldsymbol{\psi}, t) = \boldsymbol{\psi}^{\mathsf{T}}\boldsymbol{\Sigma}\boldsymbol{\psi} + \lambda E\{w(1 - \tilde{Y}[\boldsymbol{\psi}^{\mathsf{T}}\boldsymbol{X} - t])^{+}\}. \tag{7}$$

Since $w > 0$ then $E\{w(1 - \tilde{Y}[\boldsymbol{\psi}^{\mathsf{T}}\boldsymbol{X} - t])^{+}\} = E\{w(1 - \tilde{Y}[\boldsymbol{\psi}^{\mathsf{T}}\boldsymbol{X} - t])\}^{+}$. Thus, the population version in (7) is equivalent to:

$$L_{AW}(\boldsymbol{\psi}, t) = \boldsymbol{\psi}^{\mathsf{T}}\boldsymbol{\Sigma}\boldsymbol{\psi} + \lambda E\{w(1 - \tilde{Y}[\boldsymbol{\psi}^{\mathsf{T}}\boldsymbol{X} - t])\}^{+}. \tag{8}$$

Now note that:

$$E\{w(1 - \tilde{Y}[\boldsymbol{\psi}^{\mathsf{T}}\boldsymbol{X} - t])\}^{+} = E\{E\{w(1 - \tilde{Y}[\boldsymbol{\psi}^{\mathsf{T}}\boldsymbol{X} - t])\}^{+}|Y, \boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{X}]\}.$$

Since the function $a \mapsto a^+$ is convex, by Jensen's inequality we have

$$\begin{aligned} E\{[w(1-\tilde{Y}(\boldsymbol{\psi}^\intercal \boldsymbol{X}-t))]^+|Y,\boldsymbol{\beta}^\intercal \boldsymbol{X}\} \geq & \{E[w(1-\tilde{Y}(\boldsymbol{\psi}^\intercal \boldsymbol{X}-t))|Y,\boldsymbol{\beta}^\intercal \boldsymbol{X}]\}^+ \\ = & w[1-\tilde{Y}(E(\boldsymbol{\psi}^\intercal \boldsymbol{X}|\boldsymbol{\beta}^\intercal \boldsymbol{X})-t)]^+, \end{aligned}$$

where the equality follows from $Y \perp\!\!\!\perp \boldsymbol{X}|\boldsymbol{\beta}^\intercal \boldsymbol{X}$. Now we can use the following

$$E\{w[1-\tilde{Y}(\boldsymbol{\psi}^\intercal \boldsymbol{X}-t)]\}^+ \geq E\{w(1-\tilde{Y}[E(\boldsymbol{\psi}^\intercal \boldsymbol{X}|\boldsymbol{\beta}^\intercal \boldsymbol{X})-t])\}^+. \tag{9}$$

Also, note that

$$\operatorname{var}(\boldsymbol{\psi}^\intercal \boldsymbol{X}) = \operatorname{var}[E(\boldsymbol{\psi}^\intercal \boldsymbol{X}|\boldsymbol{\beta}^\intercal \boldsymbol{X})] + E[\operatorname{var}(\boldsymbol{\psi}^\intercal \boldsymbol{X}|\boldsymbol{\beta}^\intercal \boldsymbol{X})] \geq \operatorname{var}[E(\boldsymbol{\psi}^\intercal \boldsymbol{X}|\boldsymbol{\beta}^\intercal \boldsymbol{X})]. \tag{10}$$

Combining (9) and (10), we see that

$$L(\boldsymbol{\psi},t) \geq \operatorname{var}[E(\boldsymbol{\psi}^\intercal \boldsymbol{X}|\boldsymbol{\beta}^\intercal \boldsymbol{X})] + \lambda E\{w(1-\tilde{Y}[E(\boldsymbol{\psi}^\intercal \boldsymbol{X}|\boldsymbol{\beta}^\intercal \boldsymbol{X})-t])\}^+. \tag{11}$$

Note that $E(\boldsymbol{\psi}^\intercal \boldsymbol{X}|\boldsymbol{\beta}^\intercal \boldsymbol{X}) = \boldsymbol{\psi}^\intercal \boldsymbol{P}_{\boldsymbol{\beta}}^\intercal(\boldsymbol{\Sigma})\boldsymbol{X}$ where $(\boldsymbol{P}_{\boldsymbol{\beta}}(\boldsymbol{\Sigma})$ is the projection matrix $\boldsymbol{\beta}(\boldsymbol{\beta}^\intercal\boldsymbol{\Sigma}\boldsymbol{\beta})^{-1}\boldsymbol{\beta}^\intercal\boldsymbol{\Sigma})$ which implies that the right hand side of (11) is simply $L(\boldsymbol{P}_{\boldsymbol{\beta}}(\boldsymbol{\Sigma})\,\boldsymbol{\psi},t)$. That is, for every $\boldsymbol{\psi} \in \mathbb{R}^p$,

$$L(\boldsymbol{\psi},t) \geq L(\boldsymbol{P}_{\boldsymbol{\beta}}(\boldsymbol{\Sigma})\,\boldsymbol{\psi},t). \tag{12}$$

If $\boldsymbol{\psi}$ does not belong to $\mathcal{S}_{Y|\boldsymbol{X}}$, then $\operatorname{var}(\boldsymbol{\psi}^\intercal \boldsymbol{X}|\boldsymbol{\eta}^\intercal \boldsymbol{X}) > 0$, and the inequality in (10) become strict. Hence the inequality in (12) is strict. Therefore, such $\boldsymbol{\psi}$ cannot be the minimizer of $L(\boldsymbol{\psi},t)$. □

3.2 Adaptively Weighted Principal L2SVM

When one introduces weights in the Principal L2SVM algorithm then the objective function takes the following form:

$$\Lambda_{AWL2}(\boldsymbol{\psi},t) = \boldsymbol{\psi}^\intercal\boldsymbol{\Sigma}\boldsymbol{\psi} + \lambda E\{w[(1-\tilde{Y}[\boldsymbol{\psi}^\intercal(\boldsymbol{X}-E\boldsymbol{X})-t])^+]^2\}. \tag{13}$$

It is then easy to prove the following theorem.

Theorem 2 *Suppose $E(\boldsymbol{X}|\boldsymbol{\beta}^\intercal \boldsymbol{X})$ is a linear function of $\boldsymbol{\beta}^\intercal \boldsymbol{X}$, where $\boldsymbol{\beta}$ is as defined in (1). If $(\boldsymbol{\psi}^*,t^*)$ minimizes the objective function (13) among all $(\boldsymbol{\psi},t) \in \mathbb{R}^p \times \mathbb{R}$, then $\boldsymbol{\psi}^* \in \mathcal{S}_{Y|\boldsymbol{X}}$.*

The proof is omitted as it is similar to the one for the Adaptively Weighted PSVM in the previous section as well as Theorem 1 in Artemiou and Dong (2016).

4 Estimation

In this section we discuss how we construct the estimation algorithm for the adaptively weighted algorithms proposed in the previous section.

4.1 Adaptively Weighted Principal SVM

To propose the estimation algorithm for the adaptively weighted PSVM, we first write the sample version of the objective function (6), that is:

$$\hat{L}_{AW}(\boldsymbol{\psi}, t) = \boldsymbol{\psi}^{\intercal}\boldsymbol{\Sigma}_n\boldsymbol{\psi} + \frac{\lambda}{n}\sum_{i=1}^{n} w_i\{1 - \tilde{Y}_i[\boldsymbol{\psi}^{\intercal}(\boldsymbol{X}_i - \bar{\boldsymbol{X}}) - t]\}^{+}. \tag{14}$$

Then one needs to standardize the predictors using $\boldsymbol{Z}_i = \boldsymbol{\Sigma}_n^{-1/2}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})$ and $\boldsymbol{\zeta} = \boldsymbol{\Sigma}_n^{1/2}\boldsymbol{\psi}$ and the objective function becomes:

$$\hat{L}_{AW}(\boldsymbol{\zeta}, t) = \boldsymbol{\zeta}^{\intercal}\boldsymbol{\zeta} + \frac{\lambda}{n}\sum_{i=1}^{n} w_i\{1 - \tilde{Y}_i[\boldsymbol{\zeta}^{\intercal}\boldsymbol{Z}_i - t]\}^{+}. \tag{15}$$

This looks similar to the adaptively weighted large margin classifier objective function proposed by Wu and Liu (2013) in the classification framework. We first solve (15) based on the quadratic programming problem suggested by the following Theorem and then use the minimizer $\boldsymbol{\zeta}^*$ to estimate $\boldsymbol{\psi}^* = \boldsymbol{\Sigma}_n^{-1/2}\boldsymbol{\zeta}^*$ which is the minimizer of (14). Also, note that $\odot$ is used to denote the elementwise multiplication of two vectors of the same size, i.e. for vectors $\boldsymbol{a} = (a_1, \ldots a_k)$ and $\boldsymbol{b} = (b_1, \ldots, b_k)$, then $\boldsymbol{a} \odot \boldsymbol{b} = (a_1b_1, \ldots, a_kb_k)$.

Theorem 3 *If $\boldsymbol{\zeta}^*$ minimizes the objective function in (15) over $\mathbb{R}^p$, then $\boldsymbol{\zeta}^* = \frac{1}{2}\boldsymbol{Z}^{\intercal}(\alpha \odot \tilde{y})$ where α is found by solving the quadratic programming problem:*

$$\begin{aligned} &\textit{maximize} \quad \alpha^{\intercal}\mathbf{1} - \frac{1}{4}(\alpha \odot \tilde{y})^{\intercal}\boldsymbol{Z}\boldsymbol{Z}^{\intercal}(\alpha \odot \tilde{y}) \\ &\textit{subject to} \quad \mathbf{0} < \alpha < \frac{\lambda}{n}\boldsymbol{w}, \quad (\alpha \odot \tilde{y})^{\intercal}\mathbf{1} = 0. \end{aligned} \tag{16}$$

where $\mathbf{0} = (0, \ldots, 0)$, $\mathbf{1} = (1, \ldots, 1)^{\intercal} \in \mathbb{R}^n$ and $\boldsymbol{w} = (w_1, \ldots, w_n)^{\intercal}$.

Proof. Using similar developments as in Vapnik (1998) one can show that minimizing (15) is equivalent to

$$\begin{aligned} &\text{minimizing} \quad \boldsymbol{\zeta}^{\intercal}\boldsymbol{\zeta} + \frac{\lambda}{n}\boldsymbol{w}^{\intercal}\boldsymbol{\xi} \quad \text{over} \quad (\boldsymbol{\zeta}, t, \boldsymbol{\xi}) \\ &\text{subject to} \quad \boldsymbol{\xi} \geq \mathbf{0}, \quad \boldsymbol{\xi} \geq \mathbf{1} - \tilde{y} \odot (\boldsymbol{\zeta}^{\intercal}\boldsymbol{Z} - t\mathbf{1}) \end{aligned} \tag{17}$$

where $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n)$. The Lagrangian function of this problem is

$$L(\boldsymbol{c}, t, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{\zeta}^{\mathsf{T}}\boldsymbol{\zeta} + \frac{\lambda}{n}\boldsymbol{w}^{\mathsf{T}}\boldsymbol{\xi} - \boldsymbol{\alpha}^{\mathsf{T}}[\tilde{y} \odot (\boldsymbol{\zeta}^{\mathsf{T}}\boldsymbol{Z} - t\mathbf{1}) - \mathbf{1} + \boldsymbol{\xi}] - \boldsymbol{\beta}^{\mathsf{T}}\boldsymbol{\xi}. \tag{18}$$

If $(\boldsymbol{\zeta}^*, \boldsymbol{\xi}^*, t^*)$ is a solution to problem (17) then using Karush-Kuhn-Tucker Theorem, one can show that minimizing over $(\boldsymbol{\zeta}, t, \boldsymbol{\xi})$ is similar as maximizing over $(\boldsymbol{\alpha}, \boldsymbol{\beta})$. So, differentiating with respect to $\boldsymbol{\zeta}$, t, and $\boldsymbol{\xi}$ to obtain the system of equations:

$$\begin{cases} \partial L/\partial \boldsymbol{\zeta} = 2\boldsymbol{\zeta} - \boldsymbol{Z}^{\mathsf{T}}(\boldsymbol{\alpha} \odot \tilde{y}) = \mathbf{0} \\ \partial L/\partial t = \boldsymbol{\alpha}^{\mathsf{T}}\tilde{y} = 0 \\ \partial L/\partial \boldsymbol{\xi} = \frac{\lambda}{n}\boldsymbol{w} - \boldsymbol{\alpha} - \boldsymbol{\beta} = \mathbf{0} \end{cases} \tag{19}$$

Substitute the last two equations above into (18) to obtain

$$\boldsymbol{\zeta}^{\mathsf{T}}\boldsymbol{\zeta} - \boldsymbol{\alpha}^{\mathsf{T}}(\tilde{y} \odot (\boldsymbol{\zeta}^{\mathsf{T}}\boldsymbol{Z}) - \mathbf{1}). \tag{20}$$

Now substitute the first equation in (19) ($\boldsymbol{\zeta} = \frac{1}{2}\boldsymbol{Z}^{\mathsf{T}}(\boldsymbol{\alpha} \odot \tilde{y})$) in the above:

$$\mathbf{1}^{\mathsf{T}}\boldsymbol{\alpha} - \frac{1}{4}(\boldsymbol{\alpha} \odot \tilde{y})^{\mathsf{T}}\boldsymbol{Z}\boldsymbol{Z}^{\mathsf{T}}(\boldsymbol{\alpha} \odot \tilde{y}). \tag{21}$$

Thus to minimize (18) we need to maximize (21) over the constraints

$$\begin{cases} \boldsymbol{\alpha}^{\mathsf{T}}\tilde{y} = \mathbf{0} \\ \frac{\lambda}{n}\boldsymbol{w} - \boldsymbol{\alpha} - \boldsymbol{\beta} = \mathbf{0} \end{cases} \tag{22}$$

which are equivalent to the constraints in (25). □

The above result can be then used to construct the following algorithm:

1. Compute the sample mean $\bar{\boldsymbol{X}}$ and sample variance matrix $\hat{\boldsymbol{\Sigma}} = n^{-1}\sum_{i=1}^{n}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})(\boldsymbol{X}_i - \bar{\boldsymbol{X}})^{\mathsf{T}}$ and use them to standardize the data and set weights $w_i = 1, i = 1, \ldots, n$

2. Let $q_r, r = 1, \ldots, H-1$, be $H-1$ dividing points. In the simulation section we choose them to be, the $(100 \times r/H)$th sample percentile of $\{Y_1, \ldots, Y_n\}$. For each r, let $\tilde{Y}_i^r = I(Y_i > q_r) - I(Y_i \leq q_r)$ and use Theorem 3 to find $(\hat{\boldsymbol{\zeta}}_r, \hat{t}_r)$ be the minimizer of (15) where $\tilde{Y}_i$ is replaced with with $\tilde{Y}_i^r$ and weights w_i are replaced with $w_i^r = 1$ for $i = 1, \ldots, n$. This process yields $H-1$ normal vectors $\hat{\boldsymbol{\zeta}}_1, \ldots, \hat{\boldsymbol{\zeta}}_{H-1}$.

3. Use the normal vectors to calculate $\hat{\boldsymbol{\psi}}_r = \hat{\boldsymbol{\Sigma}}^{-1/2}\hat{\boldsymbol{\zeta}}_r, r = 1, \ldots, H-1$.

4. For each dividing point q_r calculate
$$w_i^r = \frac{1}{1 + |\boldsymbol{\psi}_r^\mathsf{T}(\boldsymbol{X}_i - E\boldsymbol{X}_i) - t_r|}.$$

5. Using the weights w_i^r in the previous step repeat steps 2 and 3 to find $\hat{\boldsymbol{\psi}}_r^w, r = 1, \ldots, H-1$ (new estimate of the coefficients based on the weights).

6. Construct matrix $\hat{\boldsymbol{V}}_n = \sum_{r=1}^{H-1} \hat{\boldsymbol{\psi}}_r^w \hat{\boldsymbol{\psi}}_r^{w\mathsf{T}}$.

7. Let $\hat{\boldsymbol{v}}_1, \ldots, \hat{\boldsymbol{v}}_d$ be the eigenvectors of the matrix $\hat{\boldsymbol{V}}_n$ corresponding to its d largest eigenvalues. We use subspace spanned by $\hat{\boldsymbol{v}} = (\hat{\boldsymbol{v}}_1, \ldots, \hat{\boldsymbol{v}}_d)$ to estimate the CDRS, $\mathcal{S}_{Y|\boldsymbol{X}}$.

The above algorithm is based on the “left vs right” (LVR) idea proposed by Li, Artemiou and Li (2011). It can be easily transformed to the “one vs another” (OVA) idea proposed in the same paper.

4.2 Estimation for Adaptively Weighted Principal L2SVM

A similar argument as the one used in the previous section gives us the estimation algorithm for adaptively weighted PL2SVM. One needs to show that the following Theorem holds. We omit the details due to the similarity of the arguments. First we need to write the sample version of the objective function in (13) as:

$$\hat{L}_{AWL2}(\boldsymbol{\psi}, t) = \boldsymbol{\psi}^\mathsf{T}\boldsymbol{\Sigma}_n\boldsymbol{\psi} + \frac{\lambda}{2n}\sum_{i=1}^{n} w_i(\{1 - \tilde{Y}_i[\boldsymbol{\psi}^\mathsf{T}(\boldsymbol{X}_i - \bar{\boldsymbol{X}}) - t]\}^+)^2. \tag{23}$$

Then if we standardize the predictors using $\boldsymbol{Z}_i = \boldsymbol{\Sigma}_n^{-1/2}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})$ and $\boldsymbol{\zeta} = \boldsymbol{\Sigma}_n^{1/2}\boldsymbol{\psi}$ the objective function becomes:

$$\hat{L}_{AWL2}(\boldsymbol{\zeta}, t) = \boldsymbol{\zeta}^\mathsf{T}\boldsymbol{\zeta} + \frac{\lambda}{2n}\sum_{i=1}^{n} w_i(\{1 - \tilde{Y}_i[\boldsymbol{\zeta}^\mathsf{T}\boldsymbol{Z}_i - t]\}^+)^2. \tag{24}$$

We first solve (24) based on the quadratic programming problem suggested by the following Theorem and then use the minimizer $\boldsymbol{\zeta}^*$ to estimate $\boldsymbol{\psi}^* = \boldsymbol{\Sigma}_n^{-1/2}\boldsymbol{\zeta}^*$ which is the minimizer of (23).

Theorem 4 *If $\boldsymbol{\zeta}^*$ minimizes the objective function in (24) over $\mathbb{R}^p$, then $\boldsymbol{\zeta}^* = \frac{1}{2}\boldsymbol{Z}^\mathsf{T}(\alpha \odot \tilde{y})$ where α is found by solving the quadratic programming problem:*

$$\begin{aligned} &\textit{maximize} \quad \alpha^\mathsf{T}\mathbf{1} - \frac{1}{4}(\alpha \odot \tilde{y})^\mathsf{T}\left(\boldsymbol{Z}\boldsymbol{Z}^\mathsf{T} + \frac{2n}{\lambda}\boldsymbol{D}_{\boldsymbol{w}}^{-1}\right)(\alpha \odot \tilde{y}) \\ &\textit{subject to} \quad \mathbf{0} < \alpha, \quad (\alpha \odot \tilde{y})^\mathsf{T}\mathbf{1} = 0 \end{aligned} \tag{25}$$

where $\boldsymbol{D}_{\boldsymbol{w}}$ is the diagonal matrix that has vector $\boldsymbol{w} = (w_1, \ldots, w_n)^\mathsf{T}$.

The proof of Theorem 4 is similar to the one in the previous section and therefore it is omitted. The same can be said in the estimation algorithm where the only difference is in Step 2 where one can modify it to the following:

> Let $q_r, r = 1, \dots, H-1$, be $H-1$ dividing points. In the simulation section we choose them to be, the $(100\times r/H)$th sample percentile of $\{Y_1, \dots, Y_n\}$. For each r, let $\tilde{Y}_i^r = I(Y_i > q_r) - I(Y_i \leq q_r)$ and use Theorem 4 to find $(\hat{\boldsymbol{\zeta}}_r, \hat{t}_r)$ be the minimizer of (24) where $\tilde{Y}_i$ is replaced with with $\tilde{Y}_i^r$ and weights w_i are replaced with $w_i^r = 1$ for $i = 1, \dots, n$. This process yields $H-1$ normal vectors $\hat{\boldsymbol{\zeta}}_1, \dots, \hat{\boldsymbol{\zeta}}_{H-1}$.

4.3 Asymptotic theory

Following similar developments to Li, Artemiou and Li (2011) and Artemiou and Dong (2016), one can derive the asymptotic result of the adaptively weighted algorithms. We list here only the main results for the adaptively weighted Principal L2SVM algorithm which we list without proofs as these are similar to the ones that appear to Artemiou and Dong (2016). It is important to remind here that the weights are assumed non-random.

First of all we assume $E(\boldsymbol{X}) = \mathbf{0}$ without loss of generality and we use the notation $\boldsymbol{\theta} = (\boldsymbol{\psi}^\intercal, t)^\intercal$, $\boldsymbol{Z} = (\boldsymbol{X}^\intercal, \tilde{Y})^\intercal$, $\boldsymbol{X}^\dagger = (\boldsymbol{X}^\intercal, -1)^\intercal$ and $\boldsymbol{\Sigma}^\dagger = \text{diag}(\boldsymbol{\Sigma}, 0)$, where $\text{diag}(\mathbf{A}, \mathbf{B})$ denotes a block diagonal matrix with $\mathbf{A}$ and $\mathbf{B}$ on the block diagonals. Also note that $\lambda^\dagger = \lambda 2^{-1} w$ where w is the weight. $\Lambda(\boldsymbol{\psi}, t)$ in (13) can be rewritten as $E\{m(\boldsymbol{\theta}, \boldsymbol{Z})\}$, where

$$m(\boldsymbol{\theta}, \boldsymbol{Z}) = \boldsymbol{\theta}^\intercal \boldsymbol{\Sigma}^\dagger \boldsymbol{\theta} + \lambda^\dagger \{(1 - \boldsymbol{\theta}^\intercal \boldsymbol{X} \tilde{\boldsymbol{Y}})^+\}^2. \tag{26}$$

Comparing this with the respective expression for the asymptotics of PL2SVM in Artemiou and Dong (2016) the only difference is that $\boldsymbol{\lambda}^\dagger$ includes the weight as well. This explains why the asymptotic results for the adaptively weighted method is similar to the asymptotic results of PL2SVM.

Note that we denote with $E_n\{m(\boldsymbol{\theta}, \boldsymbol{Z})\}$ the corresponding sample version of the objective function and we define $\boldsymbol{\theta}_0$ and $\hat{\boldsymbol{\theta}}$ to be the minimizers of $E\{m(\boldsymbol{\theta}, \boldsymbol{Z})\}$ and $E_n\{m(\boldsymbol{\theta}, \boldsymbol{Z})\}$ respectively. The next theorem gives the the gradient function of the L2 objective function $E\{m(\boldsymbol{\theta}, \boldsymbol{Z})\}$. To prove it one will need the prove of Lemma 1 of Artemiou and Dong (2016).

Proposition 1 *Suppose for each $\tilde{y} \in \{-1, 1\}$, the distribution of $\boldsymbol{X}|\tilde{Y} = \tilde{y}$ is dominated by the Lebesgue measure. In addition, suppose $E(\|\boldsymbol{X}\|^2) < \infty$ and $E(\|\boldsymbol{X}\|) <$*

∞. *Let* $D_{\boldsymbol{\theta}}$ *be the* $(p{+}1)$*-dimensional column vector of differential operators* $(\partial/\partial\theta_1, \ldots, \partial/\partial\theta_{p+1})^\intercal$. *Then*

$$\begin{aligned} D_{\boldsymbol{\theta}}[E\{m(\boldsymbol{\theta}, \boldsymbol{Z})\}] =& (2\boldsymbol{\psi}^\intercal\boldsymbol{\Sigma}, 0)^\intercal \\ & -2\lambda^\dagger E\left\{\boldsymbol{X}^\dagger\tilde{Y}\left[(1-\boldsymbol{\theta}^\intercal\boldsymbol{X}^\dagger\tilde{Y})I(1-\boldsymbol{\theta}^\intercal\boldsymbol{X}^\dagger\tilde{Y}>0)\right]\right\}. \end{aligned} \tag{27}$$

The next proposition finds the Hessian matrix of $\boldsymbol{\theta}$. To prove it one will need to use Lemmas 2 and 3 in Artemiou and Dong (2016).

Proposition 2 *Suppose* $\boldsymbol{X}$ *has a convex and open support, and for each* $\tilde{y} \in \{-1, 1\}$, *the distribution of* $\boldsymbol{X}|\tilde{Y} = \tilde{y}$ *is dominated by the Lebesgue measure. Let* $f_{.|.}$ *denote the conditional probability density function. Suppose, moreover:*

1. *for any linearly independent* $\boldsymbol{\psi}, \boldsymbol{\delta} \in \mathbb{R}^p$, $\tilde{y} = -1, 1$, *and* $v, \epsilon \in \mathbb{R}$, *the function*

$$\begin{aligned} u \mapsto \tilde{y}(1-\tilde{y}(u-t)-\epsilon v)E\{\boldsymbol{X}^\dagger|\boldsymbol{\psi}^\intercal\boldsymbol{X} = u, \boldsymbol{\delta}^\intercal\boldsymbol{X} = v, \tilde{Y} = \tilde{y}\}* \\ * f_{\boldsymbol{\psi}^\intercal\boldsymbol{X}|\boldsymbol{\delta}^\intercal\boldsymbol{X}, \tilde{Y}}(u|v, \tilde{y}) \end{aligned}$$

   is continuous;

2. *for any* $i = 1, \ldots, p$, *and* $\tilde{y} = -1, 1$, *there is a nonnegative function* $c_i(v, \tilde{y})$ *with* $E\{c_i(V, \tilde{Y})|\tilde{Y}\} < \infty$ *such that*

$$\begin{aligned} \tilde{y}(1-\tilde{y}(u-t)-\epsilon v)E\{X_i|\boldsymbol{\psi}^\intercal\boldsymbol{X} = u, \boldsymbol{\delta}^\intercal\boldsymbol{X} = v, \tilde{Y} = \tilde{y}\}f_{\boldsymbol{\psi}^\intercal\boldsymbol{X}|\boldsymbol{\delta}^\intercal\boldsymbol{X}, \tilde{Y}}(u|v, \tilde{y}) \\ \leq c_i(v, \tilde{y}); \end{aligned}$$

3. *for any* $\tilde{y} = -1, 1$, *there is a nonnegative function* $c_0(v, \tilde{y})$ *with* $E\{c_0(V, \tilde{Y})|\tilde{Y}\} < \infty$ *such that* $f_{\boldsymbol{\psi}^\intercal\boldsymbol{X}|\boldsymbol{\delta}^\intercal\boldsymbol{X}, \tilde{Y}}(u|v, \tilde{y}) \leq c_0(v, \tilde{y})$.

Then the function $\boldsymbol{\theta} \mapsto D_{\boldsymbol{\theta}}[E\{m(\boldsymbol{\theta}, \boldsymbol{Z})\}]$ *is differentiable in all directions with derivative matrix*

$$\boldsymbol{H} = 2\text{diag}(\boldsymbol{\Sigma}, 0) + 2\lambda^\dagger\boldsymbol{H}^\dagger \tag{28}$$

where $\boldsymbol{H}^\dagger = \sum_{\tilde{y}=-1,1} P(\tilde{Y} = \tilde{y})E\{\boldsymbol{X}^\dagger(\boldsymbol{X}^\dagger)^\intercal I(1-\boldsymbol{\theta}^\intercal\boldsymbol{X}^\dagger\tilde{y} > 0)|\tilde{Y} = \tilde{y}\}$

The next result finds the influence function of $\hat{\boldsymbol{\theta}}$.

Theorem 5 *Suppose the conditions in Propositions 1 and 2 are satisfied. Then*

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_0 - \boldsymbol{H}^{-1}D_{\boldsymbol{\theta}_0}[E\{m(\boldsymbol{\theta}_0, \boldsymbol{Z})\}] + o_P(n^{-1/2}),$$

where $\boldsymbol{H}$ *is given in Proposition 2 and* $D_{\boldsymbol{\theta}_0}[E\{m(\boldsymbol{\theta}, \boldsymbol{Z})\}]$ *in Proposition 1.*

Now let's define some notation that will help us define the asymptotic normality of the candidate matrix $\hat{\boldsymbol{V}}$. First, for a fixed dividing point q_r, we have $\tilde{Y}^r$, $r = 1, \dots, H-1$ to be the discretized responses. Then we can define $\boldsymbol{Z}^r = (\boldsymbol{X}^\intercal, \tilde{Y}^r)$, $m(\boldsymbol{\theta}, \boldsymbol{Z}^r) = \boldsymbol{\theta}^\intercal \boldsymbol{\Sigma}^\dagger \boldsymbol{\theta} - \lambda^\dagger \{(1 - \boldsymbol{\theta}^\intercal \boldsymbol{X}^\dagger \tilde{Y}^r)^+\}^2$ and let $\boldsymbol{\theta}_{0r} = (\boldsymbol{\psi}_{0r}^\intercal t_{0r})^\intercal$ be the minimizer of $E\{m(\boldsymbol{\theta}, \boldsymbol{Z}^r)\}$ over $\boldsymbol{\theta}$. The population version of $\hat{\boldsymbol{V}}$ in the estimation algorithm is thus $\boldsymbol{V} = \sum_{r=1}^{H-1} \boldsymbol{\psi}_{0r} \boldsymbol{\psi}_{0r}^\intercal$. Let $\boldsymbol{K}_{p,p}$ be the unique matrix satisfying $\boldsymbol{K}_{p,p} \text{vec}(\boldsymbol{A}) = \text{vec}(\boldsymbol{A}^\intercal)$ for any $\boldsymbol{A} \in \mathbb{R}^{p \times p}$, $\mathbf{F}_r$ be the first r rows of $\boldsymbol{H}_r^{-1}$ with $\boldsymbol{H}_r$ the Hessian of $E\{m(\boldsymbol{\theta}, \boldsymbol{Z}^r)\}$ and denote $\boldsymbol{s}_r(\boldsymbol{\theta}, \boldsymbol{Z}^r) = \mathbf{F}_r D_{\boldsymbol{\theta}_0}[E\{m(\boldsymbol{\theta}_0, \boldsymbol{Z})\}]$. Using this notation one can define the asymptotic distribution as follows:

Theorem 6 *Suppose the conditions in Propositions 1 and 2 are satisfied. Then* $\sqrt{n}\text{vec}(\hat{\boldsymbol{V}} - \boldsymbol{V})$ *converges to multivariate normal with mean* $\mathbf{0}$ *and variance* $\boldsymbol{\Lambda}_1 \boldsymbol{\Lambda}_2 \boldsymbol{\Lambda}_1$, *were* $\boldsymbol{\Lambda}_1 = \boldsymbol{I}_{p^2} + \boldsymbol{K}_{p,p}$ *and*

$$\boldsymbol{\Lambda}_2 = \sum_{r=1}^{H-1} \sum_{i=1}^{H-1} [\boldsymbol{\psi}_{0r} \boldsymbol{\psi}_{0i}^\intercal \otimes E\{\boldsymbol{s}_r(\boldsymbol{\theta}_{0r}, \boldsymbol{Z}^r) \boldsymbol{s}_i^\intercal(\boldsymbol{\theta}_{0i}, \boldsymbol{Z}^i)\}].$$

Let $\hat{\boldsymbol{U}} = (\hat{\boldsymbol{u}}_1, \dots, \hat{\boldsymbol{u}}_d)$ be the population version of $\boldsymbol{U} = (\boldsymbol{u}_1, \dots, \boldsymbol{u}_d)$, where $\boldsymbol{u}$'s are the d leading eigenvectors of $\boldsymbol{V}$. Let $\boldsymbol{D}$ be a diagonal matrix with diagonal elements being the d leading eigenvalues of $\boldsymbol{V}$. The following corollary gives the asymptotic distribution of $\hat{\boldsymbol{U}}$ and it can be proved using Corollary 1 in Bura and Pfeiffer (2008).

Corollary 1 *Suppose the conditions in Propositions 1 and 2 are satisfied, and* $\boldsymbol{V}$ *has rank* d*. Then*

$$\sqrt{n}\,\text{vec}(\hat{\boldsymbol{U}} - \boldsymbol{U}) \xrightarrow{\mathcal{D}} N\left(\mathbf{0}, (\boldsymbol{D}^{-1}\boldsymbol{U}^\intercal \otimes \boldsymbol{I}_p) \boldsymbol{\Lambda}_1 \boldsymbol{\Lambda}_2 \boldsymbol{\Lambda}_1 (\boldsymbol{D}^{-1}\boldsymbol{U}^\intercal \otimes \boldsymbol{I}_p)\right).$$

5 Simulation

In this section we run some simulations to show the improved performance of the proposed methodology. We run simulations for the following models:

$$\begin{aligned}
&\text{Model I:}\ \ Y = X_1 + X_2 + \sigma\varepsilon, \\
&\text{Model II:}\ \ Y = X_1/[0.5 + (X_2 + 1)^2] + \sigma\varepsilon, \\
&\text{Model III:}\ \ Y = X_1(X_1 + X_2 + 1) + \sigma\varepsilon,
\end{aligned}$$

where $\boldsymbol{X} \sim N(0, \boldsymbol{I}_{p \times p}), p = 10, 20, 30, \varepsilon \sim N(0, 1)$. There are 100 simulations with sample size $n = 100$. We have the number of slices $H = 20$ (we run for $H = 50$ as well but the results were similar and therefore we omit them). Finally, $\sigma = 0.2$ and the misclassification penalty $\lambda = 1$.

To compare the performance of the algorithm we use a metric proposed by Li, Zha, Chiaromonte (2005). Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two subspaces of $\mathbb{R}^p$ and $\boldsymbol{P}_{\mathcal{S}_1}$, $\boldsymbol{P}_{\mathcal{S}_2}$ the orthogonal projections on them respectively. Then the distance is measured by the matrix norm

$$\operatorname{dist}(\mathcal{S}_1, \mathcal{S}_2) = \|\boldsymbol{P}_{\mathcal{S}_1} - \boldsymbol{P}_{\mathcal{S}_2}\|. \tag{29}$$

Using the true and estimated space as the two subspaces in the formula then the above measures the distance between them. The smaller the distance the best performance of the algorithm. In our simulations we are using the Frobenius norm.

In Table 1 we can see the results of the 4 algorithms (PSVM, PL2SVM and their adaptively weighted versions) for $H = 20$ and different values of p when there are no outliers ($c = 0$, where c denotes the number of extreme outliers) and when there are two ($c = 2$) extreme outliers in the dataset. We also included SIR in our comparisons to compare with the performance of traditional methodology in the presence of extreme outliers. Those extreme outliers are created by taking the two points with the smallest response Y. We force them then to have the largest Y in the dataset by changing the sign of Y. This ensures that these two points are constantly outliers in every left vs right (LVR) comparison we apply in the dataset. As we can see there is almost always better performance of the adaptively weighted version of the two algorithms even when there are no outliers (with the exception of the PL2SVM algorithm in model 2 but even in that case the performance is close). We see that the difference between the algorithms diminishes as p gets larger. To help the reader in visualizing the performance of the algorithm, in each scenario we put in bold the algorithm that has the best performance.

6 Real data analysis

We use the airfoil self-noise dataset in the UC Irvine Machine Learning repository (Dua and Karra Taniskidou - 2017). This is a NASA dataset, obtained from a series of aerodynamic and acoustic tests of two and three-dimensional airfoil blade sections conducted in an anechoic wind tunnel and it consists of 1503 observations with 5 predictors (Frequency, Angle of attack, Chord length, Free-stream velocity, Suction side displacement thickness) and one response variable (scaled sound pressure levels).

We run PSVM and PL2SVM and the adaptively weighted versions of them, with 5 and 20 slices. The results are similar and therefore we present the ones with 5

Table 1: Mean and standard deviation (in parentheses) of the Frobenius norm for PSVM (Li, Artemiou and Li - 2011), PL2SVM (Artemiou and Dong - 2016) and the adaptively weighted versions of the two algorithms (denoted as AWPSVM and AWPL2SVM for $H = 20$. The value of c denotes the number of extreme outliers in the 100 points

			Methods				
Models	p	c	SIR	PSVM	AWPSVM	PL2SVM	AWPL2SVM
I	10	0	**0.10 (0.027)**	0.22 (0.054)	0.19 (0.049)	0.15 (0.043)	0.15 (0.043)
		2	0.37 (0.090)	0.32 (0.091)	**0.26 (0.079)**	0.31 (0.087)	0.28 (0.081)
	20	0	**0.16 (0.041)**	0.34 (0.061)	0.30 (0.057)	0.25 (0.051)	0.25 (0.051)
		2	0.55 (0.116)	0.47 (0.089)	**0.41 (0.084)**	0.47 (0.088)	0.44 (0.087)
	30	0	**0.21 (0.044)**	0.43 (0.067)	0.39 (0.064)	0.32 (0.059)	0.32 (0.059)
		2	0.71 (0.125)	0.62 (0.102)	**0.55 (0.102)**	0.61 (0.102)	0.58 (0.102)
Ⅱ	10	0	0.93 (0.251)	0.92 (0.218)	0.87 (0.208)	**0.72 (0.165)**	0.75 (0.170)
		2	1.00 (0.241)	1.04 (0.187)	0.98 (0.190)	**0.78 (0.159)**	0.82 (0.168)
	20	0	1.24 (0.199)	1.19 (0.136)	1.15 (0.135)	**1.02 (0.137)**	1.04 (0.134)
		2	1.37 (0.185)	1.29 (0.143)	1.26 (0.144)	**1.11 (0.134)**	1.13 (0.135)
	30	0	1.44 (0.115)	1.34 (0.116)	1.31 (0.118)	**1.21 (0.131)**	1.23 (0.129)
		2	1.60 (0.116)	1.41 (0.1110	1.41 (0.114)	**1.30 (0.124)**	1.32 (0.121)
Ⅲ	10	0	1.30 (0.276)	1.17 (0.258)	1.10 (0.263)	1.05 (0.259)	**1.02 (0.249)**
		2	1.44 (0.256)	1.37 (0.224)	1.33 (0.234)	1.24 (0.269)	**1.23 (0.262)**
	20	0	1.57 (0.174)	1.45 (0.177)	1.40 (0.185)	1.41 (0.179)	**1.39 (0.186)**
		2	1.70 (0.178)	1.58 (0.148)	1.54 (0.146)	1.54 (0.163)	**1.53 (0.161)**
	30	0	1.73 (0.136)	1.63 (0.130)	1.59 (0.136)	**1.58 (0.135)**	**1.58 (0.137)**
		2	1.81 (0.098)	1.69 (0.122)	1.66 (0.128)	1.66 (0.140)	**1.65 (0.140)**

Table 2: Mean and standard deviation (in parentheses) of the distance of the direction found when imposing extreme outliers to the airfoil data from the “oracle” direction

Methods			
PSVM	AWPSVM	PL2SVM	AWPL2SVM
0.026 (0.0151)	0.016 (0.0101)	0.045 (0.0190)	0.040 (0.174)

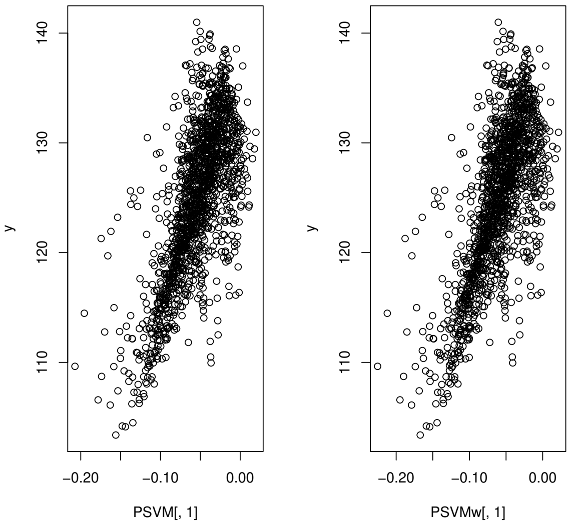

Figure 1: The picture on the left shows the first direction of the PSVM algorithm with the response and the one on the right the first direction of the adaptively weighted PSVM algorithm.

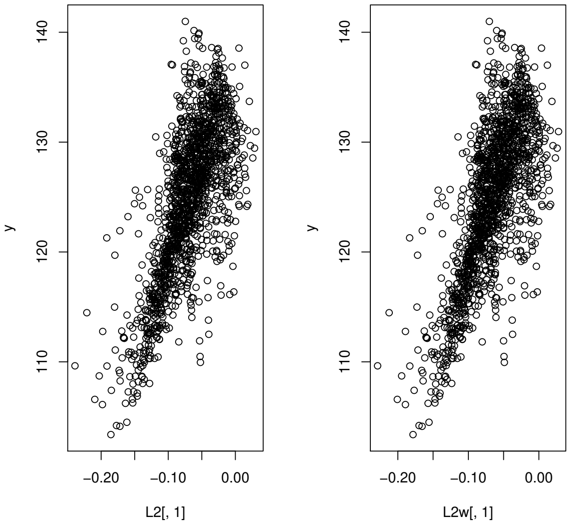

Figure 2: The picture on the left shows the first direction of the PL2SVM algorithm with the response and the one on the right the first direction of the adaptively weighted PL2SVM algorithm.

slices. All algorithms identify similar directions as is shown in Figure 1 and Figure 2. (If one looks carefully the two Figures they may be able to see that the weighted version have slightly smaller variability). The similarity of the results makes sense as it seems that there are no extreme outliers in the dataset. To demonstrate the effectiveness of the reweighted version we added 3 extreme outliers to our datasets. To create this scenario we randomly chose three out of the 10 smallest responses and change the response so that it becomes the largest. Then we measure the distance between the estimate we get when there are outliers from the true estimate we had without the outliers (essentially assuming that without the outliers we have some type of "oracle" answer). We used the Frobenius norm as with our simulations in the previous section and the results after 50 iterations are summarized in the Table 2 where we can see that there is slightly smaller distance for the adaptively weighted algorithms compared to the respective un-weighted algorithms.

7 Discussion

In this paper we present an adaptively weighted method to robustify SVM-based sufficient dimension reduction algorithms at the present of outliers. We apply a reweighting method based to the idea of Wu and Liu (2013) on the PSVM and PL2SVM algorithms proposed in Li, Artemiou and Li (2011) and Artemiou and Dong (2016) respectively. For the adaptively weighted PL2SVM we present some asymptotic results while the results for the adaptively weighted PSVM are similar to the ones presented by Shin et al (2017) and therefore are omitted. We also omitted the discussion of an order determination tests as either of the algorithms presented in Li, Artemiou and Li (2011) and Artemiou and Dong (2016) for order determination can be applied here with similar results.

We didn't discuss also the nonlinear feature extraction case of the adaptively weighted algorithms. Although one can show the theoretical developments of Li, Artemiou and Li (2011) and Artemiou and Dong (2016) to extend in the nonlinear adaptively weighted algorithms, it is not clear in the SDR framework how to calculate the weights. This has to do with the estimation procedure which calculates the sufficient predictors instead of the nonlinear hyperplane between the two classes. We believe that further investigation is needed in this case.

In the SDR literature there are some efforts to robustify inverse-moment-based dimension reduction techniques (see for example Dong et al - 2015) but to the best of our knowledge this is the first effort to robustify SVM-based sufficient dimension reduction techniques. There is scope for further investigation of these results as well as further investigation of robust algorithms in the SVM-based framework. For example, in this work we investigate what happens when we apply the weights once after running the initial algorithm with no weights. One natural extension is to apply an iterative process where the weights calculated at step i will be used to robustify the algorithm at step $i+1$. Although computationally this seems like a trivial extension, the theoretical developments will be non-trivial to develop. Furthermore, the algorithms we propose in this work reweight only points that are outliers with respect to other points in their class, that is, they are misclassified points. If an outlier is correctly classified, since it does not affect the solution then there is no reweighting. An algorithm to investigate how to robustify the original algorithms against all outliers in the dataset will, also, be interesting to explore. Finally, there is scope to investigate whether one can robustify inverse-moment-based dimension reduction techniques (like SIR (Li 1991)) by adaptively reweighting them.

References

1. Artemiou, A. and Dong, Y. (2016). Sufficient dimension reduction via principal Lq support vector machine. *Electronic Journal of Statistics*, **10**, 783–805.

2. Artemiou, A. and Shu, M. (2014). A cost based reweighed scheme of principal support vector machine. In *Topics in Nonparametric Statistics, Springer Proceedings in Mathematics & Statistics*, **74**, 1–12.

3. Bura, E. and Pfeiffer, R. (2008). On the distribution of the left singular vectors of a random matrix and its applications. *Statistics and Probability Letters*, **78**, 2275–2280.

4. Cook, R. D. (1998a). *Regression Graphics: Ideas for Studying Regressions through Graphics.* New York: Wiley.

5. Cook, R. D. (1998b). Principal Hessian directions revisited (with discussion). *Journal of the American Statistical Association*, **93**, 84–100.

6. Cook, R. D. and Weisberg, S. (1991). Discussion of "Sliced inverse regression for dimension reduction". *Journal of the American Statistical Association.* **86**, 316–342.

7. Dong, Y., Yu, Z. and Zhu, L. (2015). Robust inverse regression for dimension reduction. *Journal of Multivariate Analysis*, **134**, 71–81.

8. Li, B., Artemiou, A. and Li, L. (2011). Principal support vector machine for linear and nonlinear sufficient dimension reduction. *The Annals of Statistics*, **39**, 3182–3210

9. Li, B. and Wang, S. (2007). On directional regression for dimension reduction. *Journal of the American Statistical Association*, **102**, 997–1008.

10. Li, B., Zha, H., and Chiaromonte, F. (2005). Contour regression: a general approach to dimension reduction. *The Annals of Statistics*, **33**, 1580–1616.

11. Li, K. C. (1991). Sliced inverse regression for dimension reduction (with discussion). *Journal of the American Statistical Association*, **86**, 316–342.

12. Li, K. C. (1992). On principal Hessian directions for data visualization and dimension reduction: another application of Stein's Lemma. *Journal of the American Statistical Association*, **87**, 1025–1039.

13. Lin, Y., Lee, Y. and Wahba, G. (2002). Support vector machines for classification in nonstandard situations. *Machine Learning*, **46**, 191-202.

14. Shin, S. J., Wu, Y., Zhang, H. H. and Liu, Y. (2014). Probability-enhanced sufficient dimension reduction for binary classification. *Biometrics*, **70**, 546–555.

15. Shin, S. J., Wu, Y., Zhang, H. H. and Liu, Y. (2017). Principal weighted support vector machines for sufficient dimension reduction in binary classification. *Biometrika*, **104**, 67–81.

16. Wu, Y and Liu, Y. (2013). Adaptively weighted large margin classifiers. *Journal of Computational and Graphical Statistics*, **22**, 416–432