Iteratively-Reweighted Least-Squares Fitting of Support Vector Machines: A Majorization–Minimization Algorithm Approach

Hien D. Nguyen
Department of Mathematics and Statistics
La Trobe University
Bundoora Victoria, Australia 3086
Email: h.nguyen5@latrobe.edu.au

Geoffrey J. McLachlan
School of Mathematics and Physics
University of Queensland
St. Lucia Queensland, Australia 4072
Email: g.mclachlan@uq.edu.au

***Abstract*—Support vector machines (SVMs) are an important tool in modern data analysis. Traditionally, support vector machines have been fitted via quadratic programming, either using purpose-built or off-the-shelf algorithms. An alternative approach to SVM fitting is presented via the majorization–minimization (MM) paradigm. Algorithms that are derived via MM algorithm constructions can be shown to monotonically decrease their objectives at each iteration, as well as be globally convergent to stationary points. Constructions of iteratively-reweighted least-squares (IRLS) algorithms, via the MM paradigm, for SVM risk minimization problems involving the hinge, least-square, squared-hinge, and logistic losses, and 1-norm, 2-norm, and elastic net penalizations are presented. Successful implementations of the algorithms are demonstrated via some numerical examples.**

Keywords*—*Iteratively-reweighted least-squares; support vector machines; majorization–minimization algorithm

I. Introduction

Ever since their introduction by [1], support vector machines (SVMs) have become a mainstay in the toolkit of modern data analysts and machine learning practitioners. The popularity of SVMs is well-earned as they generally perform favorably when compared to other off-the-shelf classification techniques; for example, see [2].

Let $(\boldsymbol{X}, Y) \in \mathbb{X} \times \{-1, 1\}$ be a random observation in some probability space (consisting of a feature vector $\boldsymbol{X}$ and a classification label Y), where $\mathbb{X} \subset \mathbb{R}^p$ for $p \in \mathbb{N}$. Suppose further that $\boldsymbol{T} : \mathbb{X} \to \mathbb{T}$ is a mapping of $\boldsymbol{X}$ from $\mathbb{X}$ into some transformation space $\mathbb{T} \subset \mathbb{R}^q$ for $q \in \mathbb{N}$. For a new observation, $\boldsymbol{T}(\boldsymbol{X})$, in the transformed feature space, let $\boldsymbol{Z}^\top = \left(\boldsymbol{T}^\top(\boldsymbol{X}), Y\right)$. Here, $\top$ indicates transposition.

When constructing a soft-margin binary SVM classifier, one wishes to obtain some hyperplane $\alpha + \boldsymbol{\beta}^\top \boldsymbol{t} = 0$, such that there is a high occurrence probability of the event $Y\left[\alpha + \boldsymbol{T}^\top \boldsymbol{\beta}\right] > 0$, where $\alpha \in \mathbb{R}$, $\boldsymbol{\beta}^\top = (\beta_1, ..., \beta_q) \in \mathbb{R}^q$, and $\boldsymbol{z}^\top = (\boldsymbol{t}, y)$ is a fixed realization of $\boldsymbol{Z}$. Say that $\boldsymbol{\theta}^\top = \left(\alpha, \boldsymbol{\beta}^\top\right)$ is the parameter vector of the hyperplane.

Let $\mathbf{Z} = \{\boldsymbol{Z}_i\}_{i=1}^n$ be an IID (independent and identically distributed) sample consisting of $n \in \mathbb{N}$ observations. Under the empirical risk minimization framework of [3], the problem of obtaining an optimal hyperplane can be cast as the minimization problem

$$\hat{\boldsymbol{\theta}} = \arg\min_{\alpha, \boldsymbol{\beta}} \; R_n\left(\alpha, \boldsymbol{\beta}; \mathbf{Z}\right), \tag{1}$$

where $\hat{\boldsymbol{\theta}}^\top = \left(\hat{\alpha}, \hat{\boldsymbol{\beta}}^\top\right)$ is the optimal parameter vector. The risk R_n can be expanded into its two components $n^{-1}\sum_{i=1}^n l\left(\alpha + \boldsymbol{\beta}^\top \boldsymbol{T}_i, Y_i\right) + P(\boldsymbol{\beta})$, where $n^{-1}\sum_{i=1}^n l\left(\alpha + \boldsymbol{\beta}^\top \boldsymbol{T}_i, Y_i\right)$ is the average loss and $P(\boldsymbol{\beta})$ is a penalization function.

Under different specifications of the SVM framework, the loss function l and penalty function P can take varying forms. In the original specification (i.e. in [1]), the loss and penalty are taken to be the hinge loss function $l(w, y) = [1 - wy]_+$ and the quadratic (2-norm) penalty $P(\boldsymbol{\beta}) = \lambda \boldsymbol{\beta}^\top \boldsymbol{\beta}$, respective, where $[w]_+ = \max\{0, w\}$ for $w \in \mathbb{R}$, and $\lambda > 0$ is some penalization constant. Alternatively, [4] suggested the use of the least-squares criterion $l(w, y) = (1 - wy)^2$, instead. Some other loss functions include the squared-hinge loss $l(w, y) = [1 - wy]_+^2$, and logistic loss $l(w, y) = \log[1 + \exp(-wy)]$ (cf. [5]).

Similarly, various alternatives to the quadratic penalty have been suggested. For example [6] considered the LASSO (1-norm) penalty $P(\boldsymbol{\beta}) = \mu \sum_{j=1}^q |\beta_j|$, where $\mu > 0$ is a penalization constant. Another alternative is the elastic net-type penalty $P(\boldsymbol{\beta}) = \lambda \boldsymbol{\beta}^\top \boldsymbol{\beta} + \mu \sum_{j=1}^q |\beta_j|$ of [7].

The conventional methodology for computing (1) is to state the problem as a quadratic program, and then to solve said program via some off-the-shelf or purpose-built algorithm. See, for example, the setup in Appendix 1 of [1] and the expanded exposition of [8, Ch. 5].

As an alternative to quadratic programming, [9] derived an iteratively-reweighted least-squares (IRLS) algorithm for computing (1) under the original choices of loss and penalty of [1] via careful manipulation of the KKT conditions (Karush-Kuhn-Tucker; cf [10, Ch. 16]). Using the obtained IRLS algorithm, [11] proposed a modification for the distributed fitting of SVMs across a network.

In [12], an attempt to generalize the result of [9] to arbitrary loss functions was proposed. However, due to the wide range of loss functions that was considered in the article (e.g.

some losses were not convex), [12] were required to include a line search step in their proposed IRLS-type algorithm. Further, they were not able to provide any global convergence guarantees. The attention of this article will be restricted to the construction of IRLS algorithms for computing (1), where the risk consists of losses and penalties that are of the forms discussed above. Note that all of the losses and penalties are convex, and thus the scope is less ambitious, although more manageable than that of [12].

Using the MM (majorization–minimization) paradigm of [13] (see also [14]) and following suggestions fro [15] and [16], an IRLS algorithm for computing (1) under the specification of [1] was proposed in [17]. In this article, generalizations of the result of [17] are obtained by showing that one can compute (1) for risk functions that consist of any combination of losses and penalties from above via an IRLS algorithm by invoking the MM paradigm. Furthermore, it is shown that the constructed IRLS algorithms are all monotonic and globally convergent in the sense that the risk at every iteration is decreasing and that the iterates approach the global minimum of the risk. The interested reader is referred to [13] for a short tutorial on MM algorithms, and to [18] and [19] for an engineering perspective.

The algorithms that are presented are illustrative of the MM paradigm and it is not suggested that the results that are presented, in their current forms, are direct substitutes for the methods used in state-of-the-art solvers such as those of [20], [21], [22], and [23]; see also [24]. However, the new methods present an interesting perspective on the estimation of SVM classifiers that may be useful in combinations of losses and penalties where no current best-practice methods exist. Furthermore, further development and research into optimal implementations of MM-based methods, as well as hybridizing MM methods with current approaches, may yield computational gains on the current state-of-the-art.

The article proceeds with an introduction to MM algorithms in Section 2. MM algorithms are constructed in Section 3. Numerical examples are presented in Section 4. Finally, conclusions are drawn in Section 5.

II. Introduction to MM Algorithms

Let $F(\boldsymbol{u})$ be some objective function of interest that one wishes to minimize, where $\boldsymbol{u} \in \mathbb{U} \subset \mathbb{R}^r$ for some $r \in \mathbb{N}$. Suppose, however, that F is difficult to manipulate (e.g. non-differentiable or multimodal). Let $M(\boldsymbol{u};\boldsymbol{v})$ be defined as a majorizer of F at $\boldsymbol{v} \in \mathbb{U}$ if (i) $M(\boldsymbol{u};\boldsymbol{u}) = F(\boldsymbol{u})$ for all $\boldsymbol{u}$, and (ii) $M(\boldsymbol{u};\boldsymbol{v}) \geq F(\boldsymbol{u})$ for $\boldsymbol{u} \neq \boldsymbol{v}$.

Starting from some initial value $\boldsymbol{u}^{(0)}$, say that $\left\{\boldsymbol{u}^{(k)}\right\}$ is a sequence of MM algorithm iterates for minimizing F if it satisfies the condition

$$\boldsymbol{u}^{(k+1)} = \arg\min_{\boldsymbol{u}\in\mathbb{U}} M\left(\boldsymbol{u};\boldsymbol{u}^{(k)}\right), \tag{2}$$

for each $k \in \mathbb{N}\cup\{0\}$. The definition (2) guarantees the monotonicity of any MM algorithm.

Proposition 1. *Starting from some $\boldsymbol{u}^{(0)}$, if $\left\{\boldsymbol{u}^{(k)}\right\}$ is a sequence of MM algorithm iterates, then $\left\{F\left(\boldsymbol{u}^{(k)}\right)\right\}$ is monotonically decreasing.*

Proof: For any k, the definition of a majorizer implies that

$$\begin{aligned} F\left(\boldsymbol{u}^{(k)}\right) &= M\left(\boldsymbol{u}^{(k)};\boldsymbol{u}^{(k)}\right)\\ &\geq M\left(\boldsymbol{u}^{(k+1)};\boldsymbol{u}^{(k)}\right)\\ &\geq F\left(\boldsymbol{u}^{(k+1)}\right), \end{aligned}$$

where the second line follows from (2).

Define the directional derivative of F at $\boldsymbol{u}$ in the direction $\boldsymbol{\delta}$ as

$$F'(\boldsymbol{u};\boldsymbol{\delta}) = \lim_{\lambda\downarrow 0} \frac{F(\boldsymbol{u}+\lambda\boldsymbol{\delta}) - F(\boldsymbol{u})}{\lambda},$$

and define a stationary point of F to be any point $\boldsymbol{u}^*$ that satisfies the condition $F'(\boldsymbol{u}^*;\boldsymbol{\delta}) \geq 0$, for all $\boldsymbol{\delta}$ such that $\boldsymbol{u}+\boldsymbol{\delta}$ is a valid input of F.

In addition to the monotonicity property, it is known that MM algorithms are in globally convergent, in general, under some generous conditions. Let $\boldsymbol{u}^{(\infty)} = \lim_{k\to\infty}\boldsymbol{u}^{(k)}$ be the limit point of the MM algorithm, for some initial value $\boldsymbol{u}^{(0)}$. The following result is from [18].

Proposition 2. *Make the assumption that $M'(\boldsymbol{u},\boldsymbol{v};\boldsymbol{\delta})\,|_{\boldsymbol{u}=\boldsymbol{v}} = F'(\boldsymbol{v};\boldsymbol{\delta})$ for all $\boldsymbol{\delta}$ such that $\boldsymbol{u}+\boldsymbol{\delta}$ is a valid input of both M and F, and that $M(\boldsymbol{u},\boldsymbol{v})$ is continuous in both coordinates. Starting from some $\boldsymbol{u}^{(0)}$, if $\boldsymbol{u}^{(\infty)}$ is the limit point of the MM algorithm iterates $\left\{\boldsymbol{u}^{(k)}\right\}$, then $\boldsymbol{u}^{(\infty)}$ is a stationary point of the objective function F. Further, if F is convex, then $\boldsymbol{u}^{(\infty)}$ is a global minimum of F.*

There are numerous ways to construct majorizers; see the comprehensive treatment in [14, Ch. 4]. For the purpose of the exposition, only the following results are required.

Lemma 3. *Let $F(u) = f(u)$ be a twice differentiable function. If the second derivative satisfies $\Delta \geq f''(u)$ for all $u \in \mathbb{U}$, then F is majorized at v by*

$$M(u;v) = f(v) + f'(v)(u-v) + \frac{\Delta}{2}(u-v)^2.$$

Lemma 4. *For any $a \in [1,2]$, if $v \neq 0$, then $F(u) = |u|^a$ can be majorized at v by*

$$M(u;v) = \frac{a}{2}|v|^{a-2}u^2 + \left(1-\frac{a}{2}\right)|v|^a.$$

Corollary 5. *If $v \neq 0$, then the function $F(u) = [u]_+$ is majorized at v by*

$$M(u;v) = \frac{1}{4|v|}(u+|v|)^2.$$

Proof: Start with the identity $\max\{a,b\} = |a-b|/2 + a/2 + b/2$ and substitute $a = u$ and $b = 0$ to get $F(u) = \max\{u,0\} = [u]_+ = |u|/2 + u/2$. Apply Lemma 4 to $|u|$ (i.e. $a = 1$) to get $M(u;v) = (u^2/2|v| + |v|/2)/2 + u/2 = (u+|v|)^2/4|v|$, as required.

Lemma 3 is due to [25] and Lemma 4 arises from the derivations in [26]. It is now possible to derive the necessary majorizers for the construction of the IRLS algorithms.

III. Construction of MM Algorithms

A. Derivations of Majorizers

Begin by majorizing the mentioned loss functions from the introduction. Denote the loss functions by H (hinge loss), LS (least-square), S (squared-hinge), and L (logistic). Note that the parameter in each of the loss functions is w.

Corollary 5 can be applied directly to the hinge loss $l_H(w,y) = [1-wy]_+$ to obtain a majorizer at v:

$$M_H(w;v) = \frac{1}{4|1-vy|}(1-wy+|1-vy|)^2.$$

Just as simply, note that the least-squares loss $l_{LS}(w,y) = (1-wy)^2$ is majorized by itself; that is

$$M_{LS}(w;v) = (1-wy)^2,$$

for any v.

In order to derive a majorizer for the squared-hinge loss $l_S(w,y) = [1-wy]_+^2$, start by writing $[u]_+^2 = (|u|/2+u/2)^2 = u^2/2 + u|u|/2$. Noting that this simply implies that the function is identical to u^2 when $u \geq 0$, and 0 otherwise, one can majorize the function for any v in the two separate domains of behavior via the joint function

$$M(u;v) = u^2 I(v \geq 0) + (u-v)^2 I(v<0).$$

One can obtain the desired majorizer for l_S by setting $M_S(w;v) = M(1-wy;1-vy)$, using the expression above. Here, $I(A)$ is the indicator function, which takes value 1 if proposition A is true, and 0 otherwise.

Finally, to majorize the logistic loss $l_L(w,y) = \log[1+\exp(-wy)]$, note that the function $F(u) = \log[1+\exp(-u)]$ has first and second derivatives $F'(u) = -\pi(u)$ and $F''(u) = \pi(u)[1-\pi(u)]$, where $\pi(u) = \exp(-x)/[1+\exp(-x)]$. Now note that $0<\pi(u)<1$ and thus $F''(u) \leq 1/4$, by an elementary quadratic maximization. Thus, one can set $\Delta = 1/4$ in Lemma 3 and majorize F at v by

$$M(u;v) = F(v) - \pi(v)(u-v) + \frac{(u-v)^2}{8}.$$

One can obtain the desired majorizer for l_L by setting $M_L(w;v) = M(1-wy;1-vy)$, using the expression above.

Moving onto majorizing the penalty functions P, denote the penalties $L2$ (2-norm), $L1$ (1-norm), and E (elastic net). Starting with the 2-norm, as with the least-squares loss, the penalty is a majorizer of itself at every $\boldsymbol{v} \in \mathbb{R}^q$. That is, $P_{L2}(\boldsymbol{\beta}) = \lambda\boldsymbol{\beta}^\top\boldsymbol{\beta}$ is majorized at any $\boldsymbol{v}$ by $M_{L2}(\boldsymbol{\beta};\boldsymbol{v}) = \lambda\boldsymbol{\beta}^\top\boldsymbol{\beta} = \lambda\boldsymbol{\theta}^\top\bar{\mathbf{I}}\boldsymbol{\theta}$, where $\bar{\mathbf{I}} = \mathrm{diag}(0,1,...,1)$.

Next, the 1-norm penalty $P_{L1}(\boldsymbol{\beta}) = \mu\sum_{j=1}^q|\beta_j|$ can be majorized by application of Lemma 2 ($a=1$) for each $j\in[q]$ ($[q]=\{1,...,q\}$). That is, for each $j\in[q]$, one can majorize $|\beta_j|$ at $v\neq 0$ by $M_j(\beta_j;v_j) = \beta_j^2/2|v_j| + |v_j|/2$. Thus, one can write the majorizer of P_{L1} as

$$M_{L1}(\boldsymbol{\beta};\boldsymbol{v}) = \frac{\mu}{2}\sum_{j=1}^q\frac{\beta_j^2}{|v_j|} + \frac{\mu}{2}\sum_{j=1}^q|v_j|.$$

In the interest of numerical stability, an approximation of M_{L1} shall be considered instead, where the denominators that are in danger of going to zero are bounded away. Let $\epsilon>0$ be a small constant (i.e. $\epsilon = 10^{-6}$); one can approximate M_{L1} by

$$M_{L1}^\epsilon(\boldsymbol{\beta};\boldsymbol{v}) = \frac{\mu}{2}\sum_{j=1}^q\frac{\beta_j^2}{\sqrt{v_j^2+\epsilon}} + \frac{\mu}{2}\sum_{j=1}^q\sqrt{v_j^2+\epsilon}.$$

For further convenience, moving forward, write M_{L1}^ϵ in matrix notation with respect to the vector $\boldsymbol{\theta}$. That is, set

$$\boldsymbol{\Omega}(\boldsymbol{v}) = \mathrm{diag}\left(0,\frac{1}{\sqrt{v_j^2+\epsilon}},...,\frac{1}{\sqrt{v_j^2+\epsilon}}\right)$$

and write

$$M_{L1}^\epsilon(\boldsymbol{\beta};\boldsymbol{v}) = \frac{\mu}{2}\boldsymbol{\theta}^\top\boldsymbol{\Omega}(\boldsymbol{v})\boldsymbol{\theta} + \frac{\mu}{2}\sum_{j=1}^q\sqrt{v_j^2+\epsilon}. \tag{3}$$

With (3) in hand, it is now a simple process of combining the majorizers for the 1-norm and 2-norm penalties to get a majorizer for the elastic net penalty $P_E(\boldsymbol{\beta}) = \lambda\boldsymbol{\beta}^\top\boldsymbol{\beta} + \mu\sum_{j=1}^q|\beta_j|$. That is, one can majorize P_E when $v_j\neq 0$ by

$$M_E(\boldsymbol{\beta};\boldsymbol{v}) = \lambda\boldsymbol{\beta}^\top\boldsymbol{\beta} + \frac{\mu}{2}\sum_{j=1}^q\frac{\beta_j^2}{|v_j|} + \frac{\mu}{2}\sum_{j=1}^q|v_j|,$$

which can be approximated by

$$M_E^\epsilon(\boldsymbol{\beta};\boldsymbol{v}) = \lambda\boldsymbol{\theta}^\top\bar{\mathbf{I}}\boldsymbol{\theta} + \frac{\mu}{2}\boldsymbol{\theta}^\top\boldsymbol{\Omega}(\boldsymbol{v})\boldsymbol{\theta} + \frac{\mu}{2}\sum_{j=1}^q\sqrt{v_j^2+\epsilon}.$$

B. IRLS Algorithms

1) Hinge Loss: One can now construct the first IRLS algorithm for fitting SVMs. Begin with the original setup of [1] (i.e. $R_n = n^{-1}l_H + P_{L2}$). Let $\mathbf{z} = \{\boldsymbol{z}_i\}_{i=1}^n$ be a fixed observation of the sample $\mathbf{Z}$. One can write the risk of the sample as

$$R_n(\alpha,\boldsymbol{\beta};\mathbf{z}) = \frac{1}{n}\sum_{i=1}^n l_H\left(\alpha+\boldsymbol{\beta}^\top\boldsymbol{t}_i,y_i\right) + P_{L2}(\boldsymbol{\beta}).$$

Majorize R_n at some $\boldsymbol{\theta}^{(k)\top} = \left(\alpha^{(k)},\boldsymbol{\beta}^{(k)\top}\right)\in\mathbb{R}^{q+1}$ by

$$M(\boldsymbol{\theta};\boldsymbol{\theta}^*) = \frac{1}{n}\sum_{i=1}^n\frac{\left[1-w(\boldsymbol{\theta};\boldsymbol{t}_i)y_i+\gamma\left(\boldsymbol{\theta}^{(k)};\boldsymbol{z}_i\right)\right]^2}{4\gamma\left(\boldsymbol{\theta}^{(k)};\boldsymbol{z}_i\right)} + \lambda\boldsymbol{\theta}^\top\bar{\mathbf{I}}\boldsymbol{\theta},$$

where $w(\boldsymbol{\theta};\boldsymbol{t}_i) = \alpha+\boldsymbol{\beta}^\top\boldsymbol{t}_i$ and $\gamma(\boldsymbol{\theta};\boldsymbol{z}_i) = |1-w(\boldsymbol{\theta};\boldsymbol{t}_i)y_i|$, for each $i\in[n]$. As with the case of M_{L1}, there is a potential division by zero here. Thus, it is prudent to approximate M by M^ϵ, where

$$M^\epsilon(\boldsymbol{\theta};\boldsymbol{\theta}^*) = \frac{1}{n}\sum_{i=1}^n\frac{\left[1-w(\boldsymbol{\theta};\boldsymbol{t}_i)y_i+\gamma^\epsilon\left(\boldsymbol{\theta}^{(k)};\boldsymbol{z}_i\right)\right]^2}{4\gamma^\epsilon\left(\boldsymbol{\theta}^{(k)};\boldsymbol{z}_i\right)} + \lambda\boldsymbol{\theta}^\top\bar{\mathbf{I}}\boldsymbol{\theta},$$

and $\gamma^{\epsilon}(\boldsymbol{\theta};\boldsymbol{z}_i)=\sqrt{(1-w(\boldsymbol{\theta};\boldsymbol{t}_i)\,y_i)^2+\epsilon}$, for a small $\epsilon>0$.

Following the derivation of [17], rearrange M^{ϵ} and write

$$\begin{aligned} M^{\epsilon}\left(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}\right) &= \frac{1}{n}\left(\boldsymbol{\gamma}^{(k)}-\mathbf{Y}\boldsymbol{\theta}\right)^{\top}\mathbf{W}^{(k)}\left(\boldsymbol{\gamma}^{(k)}-\mathbf{Y}\boldsymbol{\theta}\right) \\ &\quad +\lambda\boldsymbol{\theta}^{\top}\bar{\mathbf{I}}\boldsymbol{\theta}, \end{aligned} \tag{4}$$

where $\mathbf{Y}^{\top}\in\mathbb{R}^{(q+1)\times n}$ has rows $\boldsymbol{y}_i^{\top}=\left(y_i,y_i\boldsymbol{t}_i^{\top}\right)$,

$\boldsymbol{\gamma}^{(k)\top}=\left(\gamma^{\epsilon}(\boldsymbol{\theta}^{*};\boldsymbol{z}_1)+1,...,\gamma^{\epsilon}\left(\boldsymbol{\theta}^{(k)};\boldsymbol{z}_n\right)+1\right)$, and

$$\mathbf{W}^{(k)}=\operatorname{diag}\left(\frac{1}{4\gamma^{\epsilon}\left(\boldsymbol{\theta}^{(k)};\boldsymbol{z}_1\right)},...,\frac{1}{4\gamma^{\epsilon}\left(\boldsymbol{\theta}^{(k)};\boldsymbol{z}_n\right)}\right).$$

From (4), it is easy to see that the majorizer is in a positive-quadratic form. Thus, a global minimum can be found by solving the first-order condition (FOC) of (4) (with respect to $\boldsymbol{\theta}$) to obtain

$$\boldsymbol{\theta}^{*}=\left(\mathbf{Y}^{\top}\mathbf{W}^{(k)}\mathbf{Y}+n\lambda\bar{\mathbf{I}}\right)^{-1}\mathbf{Y}^{\top}\mathbf{W}^{(k)}\boldsymbol{\gamma}^{(k)}. \tag{5}$$

One can then use (5) to construct the IRLS algorithm $\boldsymbol{\theta}^{(k+1)}=\boldsymbol{\theta}^{*}$ for the computation of (1) in the original setup of [1].

Starting from (4) and using (3), one can write an approximate majorizer for the risk combination of hinge loss and 1-norm penalty (i.e. $R_n=n^{-1}\sum l_H+P_{L1}$) at $\boldsymbol{\theta}^{(k)}$ as

$$\begin{aligned} M^{\epsilon}\left(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}\right) &= \frac{1}{n}\left(\boldsymbol{\gamma}^{(k)}-\mathbf{Y}\boldsymbol{\theta}\right)^{\top}\mathbf{W}^{(k)}\left(\boldsymbol{\gamma}^{(k)}-\mathbf{Y}\boldsymbol{\theta}\right) \\ &\quad +\frac{\mu}{2}\boldsymbol{\theta}^{\top}\boldsymbol{\Omega}^{(k)}\boldsymbol{\theta}+c, \end{aligned} \tag{6}$$

where c is a constant that does not involve $\boldsymbol{\theta}$ and $\boldsymbol{\Omega}^{(k)}=\boldsymbol{\Omega}\left(\boldsymbol{\beta}^{(k)}\right)$. Solving the FOC of (6) yields

$$\boldsymbol{\theta}^{*}=\left(\mathbf{Y}^{\top}\mathbf{W}^{(k)}\mathbf{Y}+n\frac{\mu}{2}\boldsymbol{\Omega}^{(k)}\right)^{-1}\mathbf{Y}^{\top}\mathbf{W}^{(k)}\boldsymbol{\gamma}^{(k)}. \tag{7}$$

Thus, using (7), $\boldsymbol{\theta}^{(k+1)}=\boldsymbol{\theta}^{*}$ defines an IRLS algorithm for computing (1) in the $R_n=n^{-1}\sum l_H+P_{L1}$ case.

Lastly, for the hinge loss variants, in the case of the risk combination of hinge loss and elastic net penalty (i.e. $R_n=n^{-1}\sum l_H+P_E$), one can approximate the majorizer by

$$\begin{aligned} M^{\epsilon}\left(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}\right) &= \frac{1}{n}\left(\boldsymbol{\gamma}^{(k)}-\mathbf{Y}\boldsymbol{\theta}\right)^{\top}\mathbf{W}^{(k)}\left(\boldsymbol{\gamma}^{(k)}-\mathbf{Y}\boldsymbol{\theta}\right) \\ &\quad +\lambda\boldsymbol{\theta}^{\top}\bar{\mathbf{I}}\boldsymbol{\theta}+\frac{\mu}{2}\boldsymbol{\theta}^{\top}\boldsymbol{\Omega}^{(k)}\boldsymbol{\theta}+c. \end{aligned} \tag{8}$$

Again, solving the FOC yields the minimizer

$$\boldsymbol{\theta}^{*}=\left(\mathbf{Y}^{\top}\mathbf{W}^{(k)}\mathbf{Y}+n\lambda\bar{\mathbf{I}}+n\frac{\mu}{2}\boldsymbol{\Omega}^{(k)}\right)^{-1}\mathbf{Y}^{\top}\mathbf{W}^{(k)}\boldsymbol{\gamma}^{(k)}, \tag{9}$$

which can be used to define the IRLS algorithm $\boldsymbol{\theta}^{(k+1)}=\boldsymbol{\theta}^{*}$ for computing (1), in this case.

2) Least-Squares Loss: Consider the easier cases of combinations involving the least-squares loss l_{LS}. When combined with the 2-norm penalty, one can write the risk as

$$\begin{aligned} R_n(\alpha,\boldsymbol{\beta};\mathbf{z}) &= \frac{1}{n}\sum_{i=1}^{n}(1-w(\boldsymbol{\theta};\boldsymbol{t}_i)\,y_i)^2+\lambda\boldsymbol{\theta}^{\top}\bar{\mathbf{I}}\boldsymbol{\theta} \\ &= \frac{1}{n}(\mathbf{1}-\mathbf{Y}\boldsymbol{\theta})^{\top}(\mathbf{1}-\mathbf{Y}\boldsymbol{\theta})+\lambda\boldsymbol{\theta}^{\top}\bar{\mathbf{I}}\boldsymbol{\theta}, \end{aligned}$$

where $\mathbf{1}$ is a vector of ones. Since the risk is already in a quadratic form, there is no need for an IRLS algorithm, and (1) can be obtained by solving the FOC, which yields the solution

$$\hat{\boldsymbol{\theta}}=\left(\mathbf{Y}^{\top}\mathbf{Y}+n\lambda\bar{\mathbf{I}}\right)^{-1}\mathbf{Y}^{\top}\mathbf{1}.$$

In the cases where l_{LS} is combined with either P_{L1} or P_E, however, the IRLS schemes are necessary due to the forms of the penalties. For the risk combination $R_n=n^{-1}\sum l_{LS}+P_{L1}$, the approximate majorizer at $\boldsymbol{\theta}^{(k)}$ is

$$M^{\epsilon}\left(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}\right)=\frac{1}{n}(\mathbf{1}-\mathbf{Y}\boldsymbol{\theta})^{\top}(\mathbf{1}-\mathbf{Y}\boldsymbol{\theta})+\frac{\mu}{2}\boldsymbol{\theta}^{\top}\boldsymbol{\Omega}^{(k)}\boldsymbol{\theta}+c.$$

The solution to the FOC is

$$\boldsymbol{\theta}^{*}=\left(\mathbf{Y}^{\top}\mathbf{Y}+n\frac{\mu}{2}\boldsymbol{\Omega}^{(k)}\right)^{-1}\mathbf{Y}^{\top}\mathbf{1}, \tag{10}$$

and thus the IRLS algorithm for computing (1) is to set $\boldsymbol{\theta}^{(k+1)}=\boldsymbol{\theta}^{*}$. Similarly, the IRLS algorithm for computing (1) in the $R_n=n^{-1}\sum l_{LS}+P_E$ case is to set $\boldsymbol{\theta}^{(k+1)}=\boldsymbol{\theta}^{*}$, where

$$\boldsymbol{\theta}^{*}=\left(\mathbf{Y}^{\top}\mathbf{Y}+n\lambda\bar{\mathbf{I}}+n\frac{\mu}{2}\boldsymbol{\Omega}^{(k)}\right)^{-1}\mathbf{Y}^{\top}\mathbf{1}. \tag{11}$$

3) Squared-Hinge Loss: Next, consider the combinations involving the squared-hinge loss l_S. For the combination $R_n=n^{-1}\sum l_S+P_{L2}$, one can majorize R_n at $\boldsymbol{\theta}^{(k)}$ by $M\left(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}\right)$, which equals

$$\begin{aligned} &\frac{1}{n}\sum_{i=1}^{n}(1-w(\boldsymbol{\theta};\boldsymbol{t}_i)\,y_i)^2\left[1-\upsilon\left(\boldsymbol{\theta}^{(k)};\boldsymbol{z}_i\right)\right] \\ &+\frac{1}{n}\sum_{i=1}^{n}\left(1-w(\boldsymbol{\theta};\boldsymbol{t}_i)\,y_i-\delta\left(\boldsymbol{\theta}^{(k)};\boldsymbol{z}_i\right)\right)^2\upsilon\left(\boldsymbol{\theta}^{(k)};\boldsymbol{z}_i\right) \\ &+\lambda\boldsymbol{\theta}^{\top}\bar{\mathbf{I}}\boldsymbol{\theta}, \end{aligned}$$

where $\delta\left(\boldsymbol{\theta}^{(k)};\boldsymbol{z}_i\right)=1-w\left(\boldsymbol{\theta}^{(k)};\boldsymbol{t}_i\right)y_i$, and $\upsilon\left(\boldsymbol{\theta}^{(k)};\boldsymbol{z}_i\right)=I\left(\delta\left(\boldsymbol{\theta}^{(k)};\boldsymbol{z}_i\right)<0\right)$. One can rewrite M as

$$\begin{aligned} M\left(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}\right) &= \frac{1}{n}(\mathbf{1}-\mathbf{Y}\boldsymbol{\theta})^{\top}\left[\mathbf{I}-\boldsymbol{\Upsilon}^{(k)}\right](\mathbf{1}-\mathbf{Y}\boldsymbol{\theta}) \\ &\quad +\frac{1}{n}\left(\boldsymbol{\delta}^{(k)}-\mathbf{Y}\boldsymbol{\theta}\right)^{\top}\boldsymbol{\Upsilon}^{(k)}\left(\boldsymbol{\delta}^{(k)}-\mathbf{Y}\boldsymbol{\theta}\right) \\ &\quad +\lambda\boldsymbol{\theta}^{\top}\bar{\mathbf{I}}\boldsymbol{\theta}, \end{aligned} \tag{12}$$

where $\mathbf{I}$ is the identity matrix,

$\boldsymbol{\delta}^{(k)\top}=\left(1-\delta\left(\boldsymbol{\theta}^{(k)};\boldsymbol{z}_1\right),...,1-\delta\left(\boldsymbol{\theta}^{(k)};\boldsymbol{z}_n\right)\right)$, and

$$\boldsymbol{\Upsilon}^{(k)}=\operatorname{diag}\left(\upsilon\left(\boldsymbol{\theta}^{(k)};\boldsymbol{z}_1\right),...,\upsilon\left(\boldsymbol{\theta}^{(k)};\boldsymbol{z}_n\right)\right).$$

Solving the FOC for (12) yields the solution

$$\begin{aligned}\boldsymbol{\theta}^{*} &= \left(\mathbf{Y}^{\top}\mathbf{Y}+n\lambda\bar{\mathbf{I}}\right)^{-1}\mathbf{Y}^{\top}\left(\mathbf{I}-\boldsymbol{\Upsilon}^{(k)}\right)\mathbf{1} \\ &\quad+\left(\mathbf{Y}^{\top}\mathbf{Y}+n\lambda\bar{\mathbf{I}}\right)^{-1}\mathbf{Y}^{\top}\boldsymbol{\Upsilon}^{(k)}\boldsymbol{\delta}^{(k)}, \qquad (13)\end{aligned}$$

which can be use to define the $(k+1)$ th iteration of the IRLS algorithm $\boldsymbol{\theta}^{(k+1)}=\boldsymbol{\theta}^{*}$ for computing (1) in this case.

From (12), it is not difficult to deduce that an approximate majorizer for the $R_n=n^{-1}\sum l_S+P_{L1}$ will take the form

$$\begin{aligned}M^{\epsilon}\left(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}\right) &= \frac{1}{n}\left(\mathbf{1}-\mathbf{Y}\boldsymbol{\theta}\right)^{\top}\left[\mathbf{I}-\boldsymbol{\Upsilon}^{(k)}\right]\left(\mathbf{1}-\mathbf{Y}\boldsymbol{\theta}\right) \\ &\quad+\frac{1}{n}\left(\boldsymbol{\delta}^{(k)}-\mathbf{Y}\boldsymbol{\theta}\right)^{\top}\boldsymbol{\Upsilon}^{(k)}\left(\boldsymbol{\delta}^{(k)}-\mathbf{Y}\boldsymbol{\theta}\right) \\ &\quad+\frac{\mu}{2}\boldsymbol{\theta}^{\top}\boldsymbol{\Omega}^{(k)}\boldsymbol{\theta},\end{aligned}$$

for which one can solve the FOC to obtain the solution

$$\begin{aligned}\boldsymbol{\theta}^{*} &= \left(\mathbf{Y}^{\top}\mathbf{Y}+n\frac{\mu}{2}\boldsymbol{\Omega}^{(k)}\right)^{-1}\mathbf{Y}^{\top}\left(\mathbf{I}-\boldsymbol{\Upsilon}^{(k)}\right)\mathbf{1} \\ &\quad+\left(\mathbf{Y}^{\top}\mathbf{Y}+n\frac{\mu}{2}\boldsymbol{\Omega}^{(k)}\right)^{-1}\mathbf{Y}^{\top}\boldsymbol{\Upsilon}^{(k)}\boldsymbol{\delta}^{(k)}. \qquad (14)\end{aligned}$$

One can then define the IRLS for this case as $\boldsymbol{\theta}^{(k+1)}=\boldsymbol{\theta}^{*}$. From the previous results, one can quickly deduce that the IRLS for computing (1) in the $R_n=n^{-1}\sum l_S+P_E$ case is to take $\boldsymbol{\theta}^{(k+1)}=\boldsymbol{\theta}^{*}$, where

$$\begin{aligned}\boldsymbol{\theta}^{*} &= \left(\mathbf{Y}^{\top}\mathbf{Y}+n\lambda\bar{\mathbf{I}}+n\frac{\mu}{2}\boldsymbol{\Omega}^{(k)}\right)^{-1}\mathbf{Y}^{\top}\left(\mathbf{I}-\boldsymbol{\Upsilon}^{(k)}\right)\mathbf{1} \\ &\quad+\left(\mathbf{Y}^{\top}\mathbf{Y}+n\lambda\bar{\mathbf{I}}+n\frac{\mu}{2}\boldsymbol{\Omega}^{(k)}\right)^{-1}\mathbf{Y}^{\top}\boldsymbol{\Upsilon}^{(k)}\boldsymbol{\delta}^{(k)}. \qquad (15)\end{aligned}$$

4) Logistic Loss: Finally, consider the combinations involving the logistic loss l_L. The majorizer $M\left(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}\right)$ for the risk for the combination with the 2-norm penalty (i.e. $R_n=n^{-1}\sum l_L+P_{L2}$) can be written as

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^{n}\log\left(1+\exp\left[-w\left(\boldsymbol{\theta};\boldsymbol{t}_i\right)y_i\right]\right) \\ &-\frac{1}{n}\sum_{i=1}^{n}\pi_i^{(k)}\left[w\left(\boldsymbol{\theta};\boldsymbol{t}_i\right)y_i-w\left(\boldsymbol{\theta}^{(r)};\boldsymbol{t}_i\right)y_i\right] \\ &+\frac{1}{n}\sum_{i=1}^{n}\frac{\left[w\left(\boldsymbol{\theta};\boldsymbol{t}_i\right)y_i-w\left(\boldsymbol{\theta}^{(r)};\boldsymbol{t}_i\right)y_i\right]^2}{8}+\lambda\boldsymbol{\theta}^{\top}\bar{\mathbf{I}}\boldsymbol{\theta},\end{aligned}$$

where $\pi_i^{(k)}=\pi\left(w\left(\boldsymbol{\theta}^{(k)};\boldsymbol{t}_i\right)y_i\right)$. One can rewrite the majorizer as

$$\begin{aligned}M\left(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}\right) &= \frac{1}{8n}\left(\boldsymbol{\delta}^{(k)}-\mathbf{Y}\boldsymbol{\theta}\right)^{\top}\left(\boldsymbol{\delta}^{(k)}-\mathbf{Y}\boldsymbol{\theta}\right) \\ &\quad-\frac{1}{n}\boldsymbol{\pi}^{(k)\top}\mathbf{Y}\boldsymbol{\theta}+\lambda\boldsymbol{\theta}^{\top}\bar{\mathbf{I}}\boldsymbol{\theta}+c. \qquad (16)\end{aligned}$$

The FOC solution for (16) iS

$$\begin{aligned}\boldsymbol{\theta}^{*} &= \left(\mathbf{Y}^{\top}\mathbf{Y}+8n\lambda\bar{\mathbf{I}}\right)^{-1}\mathbf{Y}^{\top}\boldsymbol{\delta}^{(k)} \\ &\quad+4\left(\mathbf{Y}^{\top}\mathbf{Y}+8n\lambda\bar{\mathbf{I}}\right)^{-1}\mathbf{Y}^{\top}\boldsymbol{\pi}^{(k)}, \qquad (17)\end{aligned}$$

which yields the IRLS $\boldsymbol{\theta}^{(k+1)}=\boldsymbol{\theta}^{*}$ for the computation of (1).

As with the previous loss functions, one must approximate the majorizer for the case of the 1-norm penalty. Thus, an approximate majorizer for the case $R_n=n^{-1}\sum l_L+P_{L1}$ can be written as

$$\begin{aligned}M^{\epsilon}\left(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)}\right) &= \frac{1}{8n}\left(\boldsymbol{\delta}^{(k)}-\mathbf{Y}\boldsymbol{\theta}\right)^{\top}\left(\boldsymbol{\delta}^{(k)}-\mathbf{Y}\boldsymbol{\theta}\right) \\ &\quad-\frac{1}{n}\boldsymbol{\pi}^{(k)\top}\mathbf{Y}\boldsymbol{\theta}+\frac{\mu}{2}\boldsymbol{\theta}^{\top}\boldsymbol{\Omega}^{(k)}\boldsymbol{\theta}+c,\end{aligned}$$

with the corresponding FOC solution

$$\begin{aligned}\boldsymbol{\theta}^{*} &= \left(\mathbf{Y}^{\top}\mathbf{Y}+4n\mu\boldsymbol{\Omega}^{(k)}\right)^{-1}\mathbf{Y}_n^{\top}\boldsymbol{\delta}^{(k)} \\ &\quad+4\left(\mathbf{Y}^{\top}\mathbf{Y}+4n\mu\boldsymbol{\Omega}^{(k)}\right)^{-1}\mathbf{Y}^{\top}\boldsymbol{\pi}^{(k)}, \qquad (18)\end{aligned}$$

which leads to the IRLS algorithm $\boldsymbol{\theta}^{(k+1)}=\boldsymbol{\theta}^{*}$.

Lastly, from the solution (18), it is not difficult to deduce that the IRLS for computing (1) in the $R_n=n^{-1}\sum l_L+P_E$ case is to take $\boldsymbol{\theta}^{(k+1)}=\boldsymbol{\theta}^{*}$, where

$$\begin{aligned}\boldsymbol{\theta}^{*} &= \left(\mathbf{Y}^{\top}\mathbf{Y}+8n\lambda\bar{\mathbf{I}}+4n\mu\boldsymbol{\Omega}^{(k)}\right)^{-1}\mathbf{Y}^{\top}\boldsymbol{\delta}^{(k)} \qquad (19) \\ &\quad+4\left(\mathbf{Y}^{\top}\mathbf{Y}+8n\lambda\bar{\mathbf{I}}+4n\mu\boldsymbol{\Omega}^{(k)}\right)^{-1}\mathbf{Y}^{\top}\boldsymbol{\pi}^{(k)}.\end{aligned}$$

C. Some Theoretical Results

In every combination except for the least-squares loss with 2-norm penalty, the MM algorithm that is derived is an IRLS algorithm. In the cases of the squared-hinge and logistic losses, in combination with the 2-norm penalty, the derived IRLS algorithm minimizes exact majorizers of the risk functions corresponding to the respective combinations. As such, Propositions 1 and 2 along with the quadratic forms of each of the risks yield the following result.

Theorem 6. *If $\left\{\boldsymbol{\theta}^{(k)}\right\}$ is a sequence that is obtained via the IRLS algorithm defined by the iterations $\boldsymbol{\theta}^{(k+1)}=\boldsymbol{\theta}^{*}$, where $\boldsymbol{\theta}^{*}$ takes the form (13) or (17), then the sequence of risk values $\left\{R_n\left(\alpha^{(k)},\boldsymbol{\beta}^{(k)};\mathbf{z}\right)\right\}$ monotonically decreases, where R_n takes the form $n^{-1}\sum l_S+P_{L2}$ or $n^{-1}\sum l_L+P_{L2}$, respectively. Furthermore, if $\boldsymbol{\theta}^{(\infty)}$ is the limit point of either IRLS algorithms, then $\boldsymbol{\theta}^{(\infty)}$ is a global minimizer of the respective risk function.*

It may serve as a minor source of dissatisfaction that in all other combinations (i.e. combinations involving either the hinge loss or the 1-norm and elastic net penalties) approximations to majorizers of the respective risk functions must be made. As such, although practically rare, the sequences of IRLS algorithm iterates for each of the aforementioned cases are not guaranteed to monotonically decrease the respective risk functions that they were derived from. However, this is not to say that the approximations are arbitrary as shall be seen from the following basic calculus fact (cf. [14, Eqn. 4.7]).

Lemma 7. *If $F(u)=f(u)$ is a concave and differentiable function for some $u\in\mathbb{U}$, then F is majorized at any $v\in\mathbb{U}$ by*

$$M\left(u;v\right)=f\left(v\right)+f'\left(v\right)\left(u-v\right).$$

Corollary 8. *The function* $F(u) = \sqrt{u}$ *is majorized at any* $v \geq 0$ *by*

$$M(u; v) = \sqrt{v} + (u - v) / (2\sqrt{y}).$$

Applying Corollary 8 with $F(u^2 + \epsilon)$ to majorize at some $v^2 + \epsilon$, where $\epsilon > 0$, yields the majorizer

$$M(u^2 + \epsilon; v^2 + \epsilon) = \sqrt{v^2 + \epsilon} + \frac{(u^2 - v^2)}{2\sqrt{v^2 + \epsilon}}. \quad (20)$$

Note that as ϵ approaches 0, $\sqrt{u^2 + \epsilon} \to |u|$ uniformly in u. Furthermore, (20) is precisely the form of the approximations that are used when the risk function involves either the hinge loss or the 1-norm and elastic net penalties. Thus, one can approximate any occurrence of an absolute value function by $\sqrt{u^2 + \epsilon}$. As such, the following result is available.

Theorem 9. *If* $\{\boldsymbol{\theta}^{(k)}\}$ *is a sequence that is obtained via the IRLS algorithm defined by the iterations* $\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^*$, *where* $\boldsymbol{\theta}^*$ *takes the forms (7), (9), (10), (15), (14), (15), (18), or (19), then there exists an approximate risk function* $R_n^\epsilon(\alpha, \boldsymbol{\beta}; \mathbf{z})$, *for which the sequence* $\{R_n^\epsilon(\alpha^{(k)}, \boldsymbol{\beta}^{(k)}; \mathbf{z})\}$ *is monotonically decreasing, where* $R_n^\epsilon(\alpha, \boldsymbol{\beta}; \mathbf{z})$ *uniformly converges to* $R_n(\alpha, \boldsymbol{\beta}; \mathbf{z})$ *(as* ϵ *approaches 0) and* R_n *is the risk function corresponding to the combination of loss and penalty of each algorithm. Furthermore, if* $\boldsymbol{\theta}^{(\infty)}$ *is a limit point of the respective algorithm, then* $\boldsymbol{\theta}^{(\infty)}$ *is a global minimizer of the respective approximate risk* R_n^ϵ.

Remark 10. In both the cases where R_n is minorized by M_n or R_n^ϵ by M_n^ϵ, the pairs of risk and majorizer functions are continuous and differentiable, respectively (for fixed ϵ in the approximate cases). Thus, in both cases, one can apply [18, Prop. 1], which establishes the satisfaction of the Proposition 2 hypotheses for differentiable functions. The results of Theorems 6 and 9 then follow from the MM construction of the algorithms.

Remark 11. Note that approximations of objectives or majorizers are inevitable in the construction of MM algorithms for non-differentiable optimization problem. See, for example, the similar approaches that are taken in [27] and [28].

IV. Numerical Examples

To demonstrate the properties of the IRLS algorithms, a numerical study is now conducted involving a small simulation. Let $\mathbf{z}_n$ be a realization of a sample from the following process. Fix $n = 10000$, for $i \in [5000]$, set $Y_i = -1$, and simulate $\boldsymbol{X}_i \in \mathbb{R}^2$ from a spherical normal distribution with mean vector $(-1, -1)$. Similarly, for $i \in [n] \setminus [5000]$, set $Y_i = 1$, and simulate $\boldsymbol{X}_i$ from a spherical normal distribution with mean vector $(1, 1)$. No transformation is made, thus set $\boldsymbol{T}_i = \boldsymbol{X}_i$, for each $i \in [n]$.

Using the simulated realization, the algorithms (13) and (17) are ran for 50 iterations each, to fit SVMs with risk functions $R_n = n^{-1}\sum l_S + P_{L2}$ and $R_n = n^{-1}\sum l_L + P_{L2}$. The risk trajectories of the two algorithms for various values of λ are visualized in Fig. 1 and 2, respectively. The algorithms (6) and (10) are also ran for 50 iterations each, to fit SVMs with risk functions $R_n = n^{-1}\sum l_H + P_{L1}$ and $R_n = n^{-1}\sum l_{LS} + P_{L1}$. The risk trajectories of the two algorithms for various values of μ are visualized in Fig. 3 and 4, respectively.

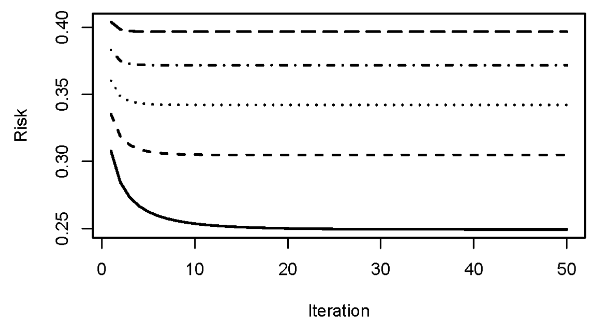

Figure 1. Risk function sequences corresponding to algorithm (13). Solid, dashed, dotted, dashed-dotted, and long dashed curves represent values $\lambda = 0, 0.1, 0.2, 0.3, 0.4$, respectively.

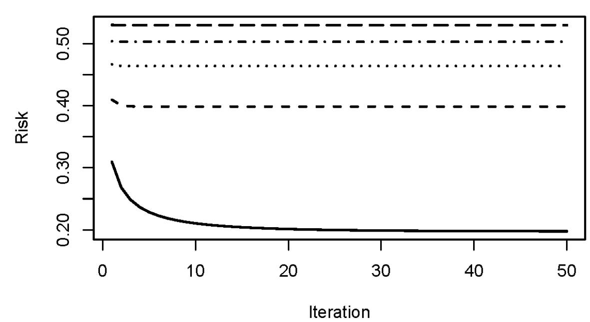

Figure 2. Risk function sequences corresponding to algorithm (17). Solid, dashed, dotted, dashed-dotted, and long dashed curves represent values $\lambda = 0, 0.1, 0.2, 0.3, 0.4$, respectively.

In order to verify that all of the algorithms are working as intended, a visualization of some separating hyperplanes is presented. Fig. 5 displays the separating hyperplanes for the four algorithms that are presented above with penalization constants set at $\lambda = 0.4$ or $\mu = 0.4$.

Observe that that the monotonicity guarantees of Theorem 6 are realized in Fig. 1 and 2. Furthermore, in Fig. 3 and 4, one can observe that even for algorithms (6) and (10), where

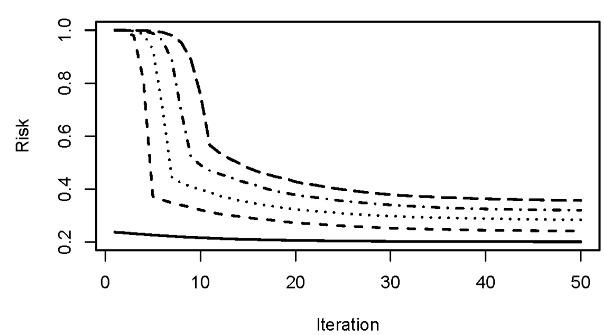

Figure 3. Risk function sequences corresponding to algorithm (6). Solid, dashed, dotted, dashed-dotted, and long dashed curves represent values $\mu = 0, 0.1, 0.2, 0.3, 0.4$, respectively.

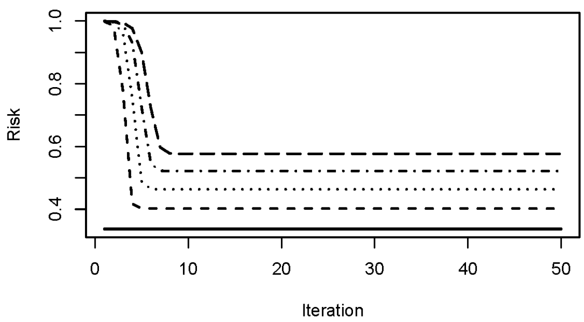

Figure 4. Risk function sequences corresponding to algorithm (10). Solid, dashed, dotted, dashed-dotted, and long dashed curves represent values $\mu = 0, 0.1, 0.2, 0.3, 0.4$, respectively.

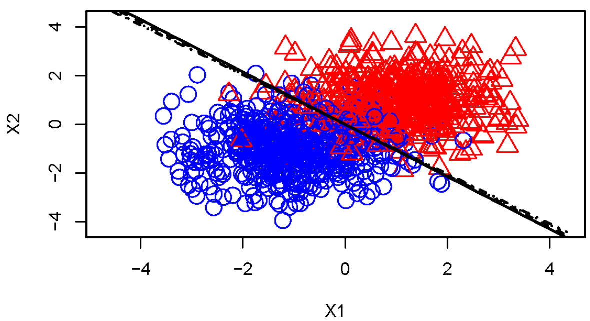

Figure 5. Separating hyperplanes obtained via four different SVM risk combinations. Solid, dashed, dotted, and dashed-dotted lines represent the hinge, least-square, squared-hinge, and logistic loss hyperplanes, respectively. Circles and triangles represent observations with label $Y_i = -1$ and $Y_i = 1$, respectively.

there are no monotonicity guarantees with respect to the actual risk (and not the approximate risk of Theorem 9), the risk trajectories are still monotonically decreasing. Thus, it can be concluded with some confidence that the approximations to the majorizers that are made in order to construct the IRLS algorithms, where necessary, do not appear to affect the performances of the IRLS algorithms with respect to the minimization of the respective risk functions. Upon inspection of Fig. 5, one can see that all four tested algorithms appear to generate hyperplanes that separate the labelled samples well.

V. Conclusions

It has been demonstrated that numerous SVM risk combinations, involving different losses and penalties, can be minimized using IRLS algorithms that are constructed via the MM paradigm. Although only four loss functions have been assessed in this article, there are numerous other currently used losses that can be minimized via MM-derived IRLS algorithms. For example, the quadratically-smoothed loss, Huber loss, and modified Huber loss of [5] as well as the Huberized-hinge loss of [29] are potential candidates for IRLS implementations.

If one wishes to undertake distributed or parallel estimation of SVMs, then the IRLS algorithms that have been presented fit easily with the distributed framework that is developed for the [1] specification by [11]. Furthermore, under appropriate probabilistic assumptions on the data generating process of $\mathbf{Z}$, the software alchemy approach of [30] for embarrassingly parallel problems can be applied in order to combine locally-computed SVM fits to yield a combined fit with the same statistical property as a fit on all of the locally-distributed data, simultaneously; see also [31].

As mentioned in the introduction, the presented methodology is currently experimental and illustrative of what is possible when one seeks to estimate SVM classifiers using MM algorithm techniques for optimization. The presented implementations are not, in their current forms, replacements for the respective state-of-the-art fitting techniques, such as those implemented in the packages referenced in [24]. However, with refinement and hybridization with current best-practice approaches, there may be avenues for potential computational gains on the current state-of-the-art via MM algorithm techniques. Furthermore, MM algorithms may also be useful for estimation of SVMs with loss and penalty combinations for which best-practice currently do not exist. One potential direction forward is the use of the developed techniques for constructing majorizers in combination within the stochastic MM algorithm framework of [32].

A minor caveat to note is the lack of ability to utilize the kernel trick for implicit feature space mapping. However, with a well-chosen transformation space $\mathbb{T}$, this should be easily resolved. Overall, the MM paradigm presents a novel perspective to algorithm construction for SVMs and has the potential to be used in interesting and productive ways.

Acknowledgements

HDN is funded by Australian Research Council grant DE170101134. GJM is funded by a Discovery Projects grant from the Australian Research Council.

References

[1] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[2] T. van Gestel, J. A. K. Suykens, B. Baesens, S. Viaene, J. Vanthenen, G. Dedene, B. de Moor, and J. Vandewalle, "Benchmarking least squares support vector machine classifiers," *Machine Learning*, vol. 54, pp. 5–32, 2004.

[3] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[4] J. A. K. Suykens and J. Vandewalle, "Least Squares Support Vector Machine," *Neural Processing Letters*, vol. 9, pp. 293–300, 1999.

[5] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *Proceedings of the twenty-first international conference on Machine learning*, 2004.

[6] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm support vector machines," in *Advances in neural information processing systems*, 2004.

[7] L. Wang, J. Zhu, and H. Zou, "The doubly regularized support vector machine," *Statistica Sinica*, vol. 16, pp. 589–615, 2006.

[8] S. Abe, *Support vector machines for pattern classification*. London: Springer, 2010.

[9] A. Navia-Vasquez, F. Perez-Cruz, A. Artes-Rodriguez, and A. R. Figueiras-Vidal, "Weighted least squares training of support vector classifiers leading to compact and adaptive schemes," *IEEE Transactions on Neural Networks*, vol. 12, pp. 1047–1059, 2001.

[10] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York: Springer, 2006.

[11] A. Navia-Vazquez, D. Gutierrez-Gonzalez, E. Parrado-Hernandez, and J. J. Navarro-Abellan, "Distributed support vector machines," *IEEE Transactions on Neural Networks*, vol. 17, pp. 1091–1097, 2006.

[12] F. Perez-Cruz, A. Navia-Vazquez, A. R. Figueiras-Vidal, and A. Artes-Rodriguez, "Empirical risk minization for support vector classifiers," *IEEE Transactions on Neural Networks*, vol. 14, pp. 296–303, 2003.

[13] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *The American Statistician*, vol. 58, pp. 30–37, 2004.

[14] K. Lange, *MM Optimization Algorithms*. Philadelphia: SIAM, 2016.

[15] J. de Leeuw and K. Lange, "Sharp quadratic majorization in one dimension," *Computational Statistics and Data Analysis*, vol. 53, pp. 2471–2484, 2009.

[16] T. T. Wu and K. Lange, "The MM alternative to EM," *Statistical Science*, vol. 25, pp. 492–505, 2010.

[17] H. D. Nguyen, "An introduction to MM algorithms for machine learning and statistical estimation," *WIREs Data Mining and Knowledge Discovery*, vol. in press, 2017.

[18] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal of Optimization*, vol. 23, pp. 1126–1153, 2013.

[19] M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang, "A unified algorithmic framework for block-structured optimization involving big data: with applications in machine learning and signal process," *IEEE Signal Processing Magazine*, vol. 33, pp. 57–77, 2016.

[20] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal Estimated sub-GrAdient SOlver for SVM," in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, 2007.

[21] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: a library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[22] V. Franc and S. Sonnenburg, "Optimal cutting plane algorithm for large-scale risk minimization," *Journal of Machine Learning Research*, vol. 10, pp. 2157–2192, 2009.

[23] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "l_p-norm multiple kernel learning," *Journal of Machine Learning Research*, vol. 12, pp. 953–997, 2011.

[24] S. Sonnenburg, G. Ratsch, S. Henschel, C. Widmer, J. Behr, A. Zien, F. de Bona, A. Binder, C. Gehl, and V. Franc, "The SHOGUN machine learning toolbox," *Journal of Machine Learning Research*, vol. 11, pp. 1799–1802, 2010.

[25] D. Bohning and B. R. Lindsay, "Monotonicity of quadratic-approximation algorithms," *Annals of the Institute of Mathematical Statistics*, vol. 40, pp. 641–663, 1988.

[26] K. Lange and J. S. Sinsheimer, "Normal/independent distributions and their applications in robust regression," *Journal of Computational and Graphical Statistics*, vol. 2, pp. 175–198, 1993.

[27] D. R. Hunter and K. Lange, "Quantile regression via an MM algorithm," *Journal of Computational and Graphical Statistics*, vol. 9, pp. 60–77, 2000.

[28] D. R. Hunter and R. Li, "Variable selection using MM algorithms," *Annals of Statistics*, vol. 33, 2005.

[29] Y. Yang and H. Zou, "An efficient algorithm for computing HHSVM and its generalizations," *Journal of Computational and Graphical Statistics*, vol. 22, pp. 396–415, 2013.

[30] N. Matloff, "Software alchemy: Turning complex statistical computations into embarrassingly-parallel ones," *Journal of Statistical Software*, vol. 71, pp. 1–15, 2016.

[31] R. Li, D. K. J. Lin, and B. Li, "Statistical inference in massive data sets," *Applied Stochastic Models in Business and Industry*, vol. 29, pp. 399–409, 2013.

[32] J. Mairal, "Stochastic majorization-minimization algorithms for large-scale optimization," in *Advances in Neural Information Processing Systems*, 2013, pp. 2283–2291.