Sparse Additive machine

Tuo Zhao **Han Liu**
Department of Biostatistics and Computer Science, Johns Hopkins University

Abstract

We develop a high dimensional nonparametric classification method named sparse additive machine (SAM), which can be viewed as a functional version of support vector machine (SVM) combined with sparse additive modeling. the SAM is related to multiple kernel learning (MKL), but is computationally more efficient and amenable to theoretical analysis. In terms of computation, we develop an efficient accelerated proximal gradient descent algorithm which is also scalable to large datasets with a provable $\mathcal{O}(1/k^2)$ convergence rate, where k is the number of iterations. In terms of theory, we provide the oracle properties of the SAM under asymptotic frameworks. Empirical results on both synthetic and real data are reported to back up our theory.

1 Introduction

The support vector machine (SVM) has been a popular classifier due to its nice computational and theoretical properties. Due to its non-smooth hinge loss, SVM possesses a robust performance (Vapnik, 1998), and the kernel trick Wahba (1999) further extends the linear SVM to more flexible nonparametric settings. However in high dimensional settings where many variables are presented but only a few of them are useful, the standard SVM suffers the curse of dimensionality and may perform poorly in practice (Hastie et al., 2009). Though many heuristic methods, such as greedy selection or recursive feature elimination (Kohavi and John, 1997; Guyon et al., 2002), have been proposed, these methods are hard to be theoretically justified. Recent development on L_1-SVM sheds some light on this problem (Wang and Shen, 2007; Bradley and Mangasarian, 1998; Zhu et al., 2003). Using the L_1-regularization, the L_1-SVM simultaneously performs variable selection and classification in high dimensions. It has been reported that L_1-SVM outperforms SVM in prediction accuracy and provide more interpretable models with fewer variables. One drawback of L_1-SVM is its linear parametric model assumption, which is restrictive in applications.

In this paper, we propose a new sparse classification method, named sparse additive machine (SAM)[1], which extends L_1-SVM to its nonparametric counterpart. By constraining the discriminant function to take an additive form, the SAM simultaneously conducts nonlinear classification and variable selection in high dimensions. Similar to the sparse additive models (SpAM) (Ravikumar et al., 2009; Liu et al., 2008), the SAM estimator is formulated as a convex optimization problem with a non-smooth objective function. The main contribution of this paper is the development of an efficient computational algorithm and an analysis of the rates of convergence in terms of the excess risk (Boyd and Vandenberghe, 2009). The algorithm is based on the recent idea of accelerated proximal gradient descent Nesterov (2005) and has a provable convergence rate of $\mathcal{O}(1/k^2)$, where k is the number of iterations. The statistical theory reveals the risk consistency (or persistency) (Greenshtein and Ritov, 2004) of the SAM even when the data dimension d is much larger than sample size n (e.g. d may increase with n almost in an exponential rate) (van der Vaart and Wellner, 2000).

There has been many related work in the literature, including the multiple kernel learning (MKL) (Bach, 2008; Christmann and Hable, 2010; Koltchinskii and Yuan, 2010; Meier et al., 2009; Lin and Zhang, 2006; Zhang, 2006). However, these methods have two drawbacks: (1) They all assume the additive function lies in a reproducing kernel Hilbert space (RKHS) and it results in an optimization problem involving nd parameters, where n is sample size and d is the dimension. This is a huge computational burden for large scale

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX. Copyright 2012 by the authors.

[1] The Significance Analysis of Microarrays is also called SAM, but it targets at a completely different problem

problems. (2) Existing theoretical analysis for high dimensional MKL require smooth loss function. On the contrast, the SAM is computationally scalable by reducing the number of parameters to approximately $\mathcal{O}(n^{1/5}d)$ and enjoys the theoretical guarantees on the non-smooth hinge loss function.

In the next section we establish necessary notation and assumptions. In Section 3 we formulate the SAM as an optimization problem and derive a scalable algorithm in Section 4. Some theoretical analysis is provided in Section 5. Section 6 presents some numerical results on both simulated and real data.

2 Notations and Assumptions

We consider a classification problem with an input variable $X = (X_1, X_2, ..., X_d)^T \in [0,1]^d$ and an output variable (or class label) $Y \in \{+1, -1\}$. Let $f : [0,1]^d \to \{-1, +1\}$ be the discriminant function and $\{(x_i, y_i)\}_{i=1}^n$ be the observed data points, we want to find a function f that minimizes the risk: $\text{Risk}(f) \equiv \mathbb{E}(L(Y, f(x))$, where L is some convex loss function.

For any integrable function $f_j : [0,1] \to \mathbb{R}$, we define its L_2-norm by

$$\|f\|_2 = \sqrt{\int f(x)^2 dx}.$$

For $j \in 1, ..., d$, let $\mathcal{H}_j$ denote the Hilbert subspace of L_2. To make the later model identifiable, we also constrain $\mathbb{E}\left(f_j(X_j)\right) = 0$. Let $\mathcal{H} = \mathcal{H}_1 \oplus \mathcal{H}_2 \oplus ... \oplus \mathcal{H}_d$ be the Hilbert space of functions of $(x_1, ..., x_d)$ that take an additive form: $f(x) = b + \sum_{j=1}^d f_j(x_j)$, with $f_j \in \mathcal{H}_j, j = 1, ..., d$. Let$\{\psi_{jk} : k = 1, 2, ...\}$ denote a uniformly bounded, orthonormal basis with respect to Lebesgue measure on $[0,1]$. Unless stated otherwise, we assume that $f_j \in \mathcal{T}_j$ where

$$\mathcal{T}_j = \Big\{f_j \in \mathcal{H}_j : f_j(x_j) = \sum_{k=0}^{\infty} \beta_{jk}\psi(x_j), \\ \sum_{k=0}^{\infty} \beta_{jk}^2 k^{v_j} \leq C\Big\} \text{ for some } 0 < C < \infty.$$

where v_j is the smoothness parameter. In the sequel, we assume $v_j = 2$ although the extension to general settings is straightforward. It is possible to adapt to v_j although we do not pursue this direction. Since we assume $\int \psi_{jk}\psi_{jl} = 0$ for any $k \neq l$, we further have

$$\|f_j\|_2 = \sqrt{\int \left(\sum_{k=1}^{\infty} \beta_{jk}\psi_{jk}(x_j)\right)^2 dx_j} = \sqrt{\sum_{k=1}^{\infty} \beta_{jk}^2}.$$

For $v = (v_1, ..., v_d)^T$, we define

$$||v||_2 = \left(\sum_{j=1}^{k} v_j^2\right)^{\frac{1}{2}} \text{ and } ||v||_1 = \sum_{j=1}^{k} |v_j|.$$

3 Sparse Additive machine

Let $L(y, f(x)) = (1 - yf(x))_+ \equiv \max(1 - yf(x), 0)$ be the hinge loss function. Consider a linear discriminant function $f(x) = b + w^T x$, the L_1-SVM takes the form

$$\min_{w,b} \frac{1}{n} \sum_{i=1}^{n} L(y_i, b + w^T x_i) + \lambda \|w\|_1,$$

where $\lambda > 0$ as a regularization parameter.

For the sparse additive machine, we no longer constrain $f(x)$ to be linear function of x. Instead, $f(x)$ is chosen as an additive forms: $f(x) = b + \sum_{j=1}^d f_j(x_j)$. The sparse additive machine can be formulated as

$$\min_{f_j \in \mathcal{T}_j, 1 \leq j \leq d} \frac{1}{n} \sum_{i=1}^{n} L(y_i, b + \sum_{j=1}^{d} f_j(x_{ij})) + \lambda \sum_{j=1}^{d} \|f_j\|_2.$$

To obtain smooth estimates, we use truncated basis estimates. Recall $\{\psi_{jk} : k = 1, 2, ...\}$ be an orthogonal basis for $\mathcal{T}_j$ and $\sup_x |\psi_{jk}(x)| \leq \kappa$ for some $\kappa \leq \infty$. Then $f_j(x_j) = \sum_{k=1}^{\infty} \beta_{jk}\psi_{jk}(x_j)$, where $\beta_{jk} = \int f_j(x_j)\psi_{jk}(x_j)dx_j$. We define $\widetilde{f}_j(x_j) = \sum_{k=1}^{p} \beta_{jk}\psi_{jk}(x_j)$ to be a smoothed approximation and $\|\widetilde{f}_j\|_2 = \sqrt{\sum_{k=1}^{p_n} \beta_{jk}^2}$ with the truncation rate $p = p_n$. It is well known that for the second order Sobolev ball $\mathcal{T}_j$ we have $\|f_j - \widetilde{f}_j\|_2^2 = O(1/p^4)$. Let $S = \{j : f_j \neq 0\}$. Assuming the sparsity condition $|S| = \mathcal{O}(1)$, it follows that $\|f - \widetilde{f}\|_2^2 = \mathcal{O}(1/p^4)$. where $\widetilde{f} = \sum_{j=1}^d \widetilde{f}_j$. The usual choice is $p \asymp n^{1/5}$ yielding $\|f - \widetilde{f}\|^2 = \mathcal{O}(n^{-4/5})$

We define

$$\Psi_i = \left(\psi_1(x_{i1}), ..., \psi_{p_n}(x_{i1}), ..., \psi_1(x_{id}), ..., \psi_{p_n}(x_{id})\right)^T,$$

where $i = 1, ..., n$ and

$$\beta = (\beta_{11}, ..., \beta_{1p_n}, ..., \beta_{d1}, ..., \beta_{dp_n})^T$$

with $\beta_j = (\beta_{j1}^T, ..., \beta_{jp_n}^T)^T, j = 1...d$. Since the constants $1/n$ in the loss term can be absorbed by the regularization parameter λ, eventually we can rewrite the equivalent form of the SAM as below

$$\min_{b, \beta_j, 1 \leq j \leq d} \sum_{i=1}^{n} L\left(y_i, b + \Psi_i^T \beta\right) + \lambda \sum_{j=1}^{d} ||\beta_j||_2. \tag{1}$$

From a computational perspective, we formulate the SAM as a unconstrained Lagrangian form (1). But it is more convenient to use an alternative constrained form (2) to analyze the theoretical properties. From the duality theory, it is straightforward to see that these two forms are one-to-one equivalent.

$$\min_{b, \beta_j, 1 \leq j \leq d} \sum_{i=1}^{n} L\left(y_i, b + \Psi_i^T \beta\right) \text{ s.t.} \sum_{j=1}^{d} \|\beta_j\|_2 \leq s. \tag{2}$$

For notational simplicity, in the rest of this paper we absorb the constant term b into β by augmenting

$\widetilde{\Psi}_i = (1, \Psi_i^T)^T$ and $\widetilde{\beta} = (b, \beta^T)^T$. We define the objective function in (2) as $F(\widetilde{\beta})$, $L_i(\widetilde{\beta}) = L(y_i, \widetilde{\Psi}_i^T \widetilde{\beta})$ and $L_*(\widetilde{\beta}) = \sum_{i=1}^n L_i(\widetilde{\beta})$, $R_j(\beta_j) = ||\beta_j||_2$ and $R_*(\widetilde{\beta}) = \sum_{j=1}^d R_j(\beta_j)$. R_* is often referred to group regularization. This convex optimization problem can be solved by simple solvers using subgradients, which is usually not efficient. We develop an efficient algorithm based on Nesterov's method Nesterov (2005) to handle the non-smooth objective function $F(\widetilde{\beta})$.

4 Computational Algorithm

The algorithm has two stages: *smooth approximation* and *gradient acceleration*. In the first stage, some duality arguments are carried on so that smooth differentiable approximations for both $L_*(\widetilde{\beta})$ and $R_*(\widetilde{\beta})$ are constructed with guaranteed precision. The approximations errors are uniformly bounded by some positive smoothing parameters. In the second stage, an acceleration trick is applied so that the first-order method can achieve the rate of second-order methods. The algorithm is iterative and within each iteration the gradient is constructed by a weighted average of current gradient and historical gradients. Previous iterations can help to adjust the descent and further achieve the optimal rate of convergence $\mathcal{O}(1/k^2)$ without tuning the step size, where k is the number of iterations.

4.1 Smooth the Hinge Loss

The hinge loss function has the following equivalent form. For any $\widetilde{\beta}$,

$$L_*(\widetilde{\beta}) = \sum_{i=1}^n L_i(\widetilde{\beta}) = \max_{u \in \mathcal{P}} \sum_{i=1}^n \left(1 - y_i \widetilde{\Psi}_i^T \widetilde{\beta}\right) u_i,$$

where $\mathcal{P} = \{u : 0 \leq u_i \leq 1, u \in \mathbb{R}^n\}$. We consider the following function

$$L_*^{\mu_1}(\widetilde{\beta}) \equiv \sum_{i=1}^n L_i^{\mu_1}(\widetilde{\beta}) \equiv \max_{u \in \mathcal{P}} \sum_{i=1}^n \left(1 - y_i \widetilde{\Psi}_i^T \widetilde{\beta}\right) u_i - d_1(u),$$

where $d_1(u) = \frac{\mu_1}{2}||u||_2^2$ is a prox-function. Since $d(u)$ is strongly convex, the maximizer u^* is unique:

$$u_i^* = \text{median}\left(0, \frac{1 - y_i \widetilde{\Psi}_i^T \widetilde{\beta}}{\mu_1}, 1\right), \forall i = 1..., n.$$

$L_*^{\mu_1}(\widetilde{\beta})$ is well defined, convex, continuously differentiable and can be seen as a uniformly smooth approximation of $L_*(\widetilde{\beta})$ and obviously for any $\widetilde{\beta}$, we have $L_*^{\mu_1}(\widetilde{\beta}) \leq L_*(\widetilde{\beta}) \leq L_*^{\mu_1}(\widetilde{\beta}) + n\mu_1$. Moreover, its gradient

$$\nabla L_*^{\mu_1}(\widetilde{\beta}) = -\sum_{i=1}^n y_i \widetilde{\Psi}_i u_i^*$$

is Lipschitz continuous with a Lipschitz constant $\mathcal{C}_{L_*^{\mu_1}} = n \max_{1 \leq i \leq n} \|\widetilde{\Psi}_i^T\|_2^2 / \mu_1$. The smoothed hinge loss $R_i^{\mu_1}$ with different μ_1's are shown in Figure 1.

4.2 Smooth the Group Regularization

Similarly, the group regularization also have the following equivalent form for any $\widetilde{\beta}$,

$$R_*(\widetilde{\beta}) = \sum_{j=1}^d R_j(\beta_j) = \max_{v_j \in \mathcal{P}} \sum_{j=1}^d v_j^T \beta_j,$$

where $\mathcal{P} = \{v_j : ||v_j|| \leq 1, v_j \in \mathbb{R}^{p_n}, j = 1, ..., d\}$. We consider the following function

$$R_*^{\mu_2}(\widetilde{\beta}) \equiv \sum_{j=1}^d R_j^{\mu_2}(\beta_j) \equiv \max_{v_j \in \mathcal{P}} \sum_{j=1}^d \left(v_j^T \beta_j - d_2(v_j)\right),$$

where $d_2(v_j) = \frac{\mu_2}{2}||v_j||_2^2$ is also a prox-function. Therefore the maximizers $v_1^*, \ldots, v_d^*$ are unique:

$$v_j^* = \frac{\beta_j}{\mu_2 \max(||\widetilde{v}_j||_2, 1)}, \forall j = 1, ..., d.$$

Similarly $R_*^{\mu_2}(\widetilde{\beta})$ is also well defined, convex, continuously differentiable and can be seen as a uniform smooth approximation of $R_*(\widetilde{\beta})$ and obviously for any $\widetilde{\beta}$, we have $R_*^{\mu_2}(\widetilde{\beta}) \leq R_*(\widetilde{\beta}) \leq R_*^{\mu_2}(\widetilde{\beta}) + d\mu_2$. Moreover, its gradient

$$\nabla R_*^{\mu_2}(\widetilde{\beta}) = \left(0, v_1^{*T}, ..., v_d^{*T}\right)^T$$

is Lipschitz continuous with a Lipschitz constant $\mathcal{C}_{R_*^{\mu_2}} = d/\mu_2$. Figure 2 plots the group regularization and the smoothed approximation with different μ_2's.

4.3 Accelerated Gradient

In the second stage, we focus on minimizing $F^\mu(\widetilde{\beta}) \equiv L_*^\mu(\widetilde{\beta}) + \lambda R_*^\mu(\widetilde{\beta})$, which is the smooth approximation of the original objective function. The gradient of $F^\mu(\widetilde{\beta})$ and corresponding Lipschitz constant are computed as

$$\nabla F^\mu = \nabla L_*^\mu + \lambda \nabla R_*^\mu \quad \text{and} \quad \mathcal{C}_\mu = \mathcal{C}_{L_*^\mu} + \lambda \mathcal{C}_{R_*^\mu}.$$

The Nesterov's method enjoys two attractive features: 1. it can achieve a convergence rate similar to 2nd order methods such as Newton, but based only on the gradient (1st order). 2. The step size can be automatically chosen by two auxiliary optimization problems without line search. In the k-th iteration of the Nesterov's method, we consider the following two optimization problems,

$$\min_{\alpha \in \mathbb{R}^{d \cdot p_n + 1}} \quad (\alpha - \widetilde{\beta}^{(k)})^T \nabla F^\mu\left(\widetilde{\beta}^{(k)}\right) + \frac{\mathcal{C}_\mu}{2}||\alpha - \widetilde{\beta}^{(k)}||_2^2,$$

$$\min_{\gamma \in \mathbb{R}^{d \cdot p_n + 1}} \quad \frac{\mathcal{C}_\mu}{2}||\gamma - \widetilde{\beta}^{(0)}||_2^2 + \sum_{t=1}^{(k)} \frac{(t+1)}{2}\Bigg(F^\mu\left(\widetilde{\beta}^{(t)}\right) + \left(\gamma - \widetilde{\beta}^{(t)}\right)^T \nabla F^\mu\left(\widetilde{\beta}^{(k)}\right)\Bigg). \qquad (3)$$

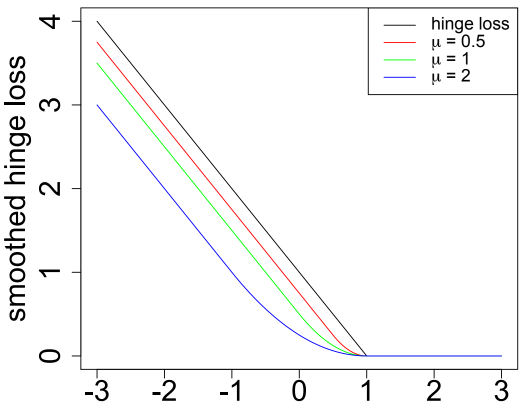
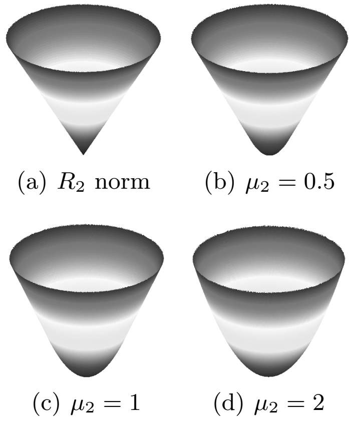


Figure 1: Smoothed hinge loss using different μ_2's



(a) R_2 norm (b) $\mu_2 = 0.5$

(c) $\mu_2 = 1$ (d) $\mu_2 = 2$

Figure 2: Smoothed Group Regularizer

With the Lipschitz constant working as a regularization parameter to avoid a radical step size, the algorithm attempts to maximize the descent. By directly setting the gradients of the two objective functions equal to zero in the auxiliary optimization problems, we can obtain $\alpha^{(k)}$, $\gamma^{(k)}$ and $\widetilde{\beta}^{(k+1)}$ respectively,

$$\begin{aligned}\alpha^{(k)} &= \widetilde{\beta}^{(k)} - \frac{\nabla F^{\mu}\left(\widetilde{\beta}^{(k)}\right)}{\mathcal{C}_{\mu}}, &(4)\\ \gamma^{(k)} &= \widetilde{\beta}^{(0)} - \sum_{t=1}^{k} \frac{t+1}{2\mathcal{C}_{\mu}} \nabla F^{\mu}\left(\widetilde{\beta}^{(t)}\right), &(5)\\ \widetilde{\beta}^{(k+1)} &= \frac{2\gamma^{(k)} + (k+1)\alpha^{(k)}}{k+3}. &(6)\end{aligned}$$

Here $\alpha^{(k)}$ is the standard gradient descent solution with step size $1/\mathcal{C}_{\mu}$ at the k-th iteration. $\gamma^{(k)}$ is a solution to a gradient decent step that starts from the initial value and proceed along a direction determined by the weighted sum of negative gradients in all previous iteration. The weights of the later gradients are larger than earlier ones. Therefore, $\widetilde{\beta}^{(k+1)}$ encodes both current gradient ($\alpha^{(k)}$) and historical gradients ($\gamma^{(k)}$) . The optimal convergence rate can be derived based on Theorem 2 in Nesterov (2005).

4.4 Convergence Analysis

Lemma 4.1 *Let $\phi^{(k)}$ be the optimal object value of the optimization* (3)*, for any k and the corresponding $\alpha^{(k)}$, $\gamma^{(k)}$ and $\beta^{(k)}$ defined in* (4)*,* (5) *and* (6)*, respectively, we have*

$$\frac{(k+1)(k+2)}{4} \nabla F^{\mu}\left(\alpha^{(k)}\right) \leq \phi^{(k)}. \qquad (7)$$

Lemma 4.1 is a direct result of Lemma 2 in Nesterov (2005) and can be applied to analyze the convergence rate of our APG algorithm.

Theorem 4.2 *The convergence rate of the APG algorithm is $\mathcal{O}(1/k^2)$. It requires $\mathcal{O}(1/\sqrt{\epsilon})$ iterations to achieve an ϵ accurate solution.*

Proof Let the optimal solution be β^*. Since $F^{\mu}(\beta)$ is a convex function, we have

$$F^{\mu}\left(\beta^*\right) \geq F^{\mu}\left(\beta^{(t)}\right) + \left(\beta^* - \beta^{(t)}\right)^T \nabla F^{\mu}\left(\beta^{(t)}\right).$$

Thus,

$$\begin{aligned}\phi^{(k)} &\leq \mathcal{C}_{\mu}\|\beta^* - \beta^{(0)}\|_2^2 + \sum_{t=1}^{(k)} \frac{(t+1)}{2} (F^{\mu}\left(\beta^{(t)}\right)\\ &\qquad + \left(\beta^* - \beta^{(t)}\right)^T \nabla F^{\mu}\left(\beta^{(k)}\right))\\ &\leq \mathcal{C}_{\mu}\|\beta^* - \beta^{(0)}\|_2^2 + \sum_{t=1}^{(k)} \frac{(t+1)}{2} F^{\mu}\left(\beta^*\right)\\ &= \mathcal{C}_{\mu}\|\beta^* - \beta^{(0)}\|_2^2 + \frac{(k+1)(k+2)}{4} \nabla F^{\mu}(\beta^*).\end{aligned}$$

According to Lemma 4.1, we have

$$\begin{aligned}&\frac{(k+1)(k+2)}{4} \nabla F^{\mu}(\alpha^{(k)})\\ &\quad \leq \phi^{(k)} \leq \mathcal{C}_{\mu}\|\beta^* - \beta^{(0)}\|_2^2 + \frac{(k+1)(k+2)}{4} \nabla F^{\mu}(\beta^*).\end{aligned}$$

Hence the accuracy at the k-th iteration is

$$\nabla F^{\mu}(\alpha^{(k)}) - \nabla F^{\mu}(\beta^*) \leq \frac{4\mathcal{C}_{\mu}\|\beta^* - \beta^{(0)}\|_2^2}{(k+1)(k+2)}.$$

Therefore, APG converges at rate $\mathcal{O}(1/k^2)$, and the minimum iteration number to reach an ϵ accurate solution is $\mathcal{O}(1/\sqrt{\epsilon})$.

5 Theoretical Properties

We analyze the asymptotic properties of the SAM in high-dimensions, where d is allowed to grow with n at a speed no faster than $\exp(n/p)$. For simplicity, we assume $X_j \in [0, 1]$ for $1 \leq j \leq d$. We define

$$\mathcal{F}(d,p,s) \equiv \Big\{f : [0,1]^d \to \mathbb{R};\ \text{where } f(x) = \sum_{j=1}^{d} f_j(x_j),$$

$$f_j(x_j) = \sum_{k=0}^{\infty}\beta_{jk}\psi(x_j), \sum_{j=1}^{d}\sqrt{p\sum_{k=1}^{p}\beta_{jk}^2 \le s}\Big\}.$$

We denote $\mathcal{F}(d,p) = \cup_{0\le s<\infty}\mathcal{F}(d,p,s)$ to be the full d-dimensional model.

Let $f^{(d,p)} = \operatorname{arginf}_{f\in\mathcal{F}(d,p)} \mathbb{E}L(Y,f)$, where $f^{(d,p)}$ may not belong to $\mathcal{F}(d,p)$. For any $f \in \mathcal{F}(d,p)$, we define the excess hinge risk to be

$$\text{Risk}\left(f, f^{(d,p)}\right) \equiv \mathbb{E}L(Y,f(X)) - \mathbb{E}L(Y,f^{(d,p)}).$$

The following theorem yields a rate of $\text{Risk}\left(\widehat{f}, f^{(d,p)}\right)$, when $d = d_n$, $s = s_n$ and $p = p_n$ grow with n, as $n \to \infty$.

Theorem 5.1 *Assume that $p_n \log d = o(n)$. Let*

$$\widehat{f} = \widehat{b} + \sum_{j=1}^{d}\sum_{k=1}^{p}\widehat{\beta}_{jk}\psi(x_j),$$

where $\widehat{b}$ and $\widehat{\beta}_{jk}$ are solutions to Eq. (2). We have the following oracle inequality.

$$\text{Risk}\left(\widehat{f}, f^{(d,p)}\right) = O_P\left(\eta + s\sqrt{\frac{p\log d}{n}}\right), \qquad (8)$$

where $\eta = \inf_{f\in\mathcal{F}(d,p,s)} \text{Risk}\left(f, f^{(d,p)}\right)$.

If $f^{(d,p)} \in \mathcal{F}(d,p,s)$ and $p \asymp n^{1/5}$, we have

$$\text{Risk}\left(\widehat{f}, f^{(d,p)}\right) = O_P\left(s\sqrt{\frac{\log d}{n^{4/5}}}\right).$$

This rate is optimal up to a logarithmic term.

Proof We only provide a proof sketch due to the space limit. Recall we have $\sup_x|\psi_{jk}| \le \kappa$, similar to the normalization condition in Lasso and group Lasso Liu and Zhang (2009), we require $\kappa \le \frac{1}{\sqrt{p}}$. We first show that the estimated $\widehat{b}$ is a bounded quantity. To see this, since $\widehat{b}$ and $\widehat{\beta}$ minimizes (1), we have $\max_i(1 - y_i(\widehat{b} + \Psi_i^T\widehat{\beta})) \ge 0$ leading to

$$|\widehat{b}| \le \frac{\|\widehat{\beta}\|_1}{\sqrt{p}} + 1 \le \sum_{j=1}^{d}\|\widehat{\beta}_j\|_2 + 1 \le s + 1.$$

Thus $\widehat{f} \in \mathcal{F}^b(d,p,s) \equiv \mathcal{F}(d,p,s) \cap \{f : |b| \le s+1\}$.

For simplicity, we use the notation $Z = (X,Y)$ and $z_i = (x_i, y_i)$. Let P denote the distribution Z and $f_0 = \operatorname{argmin}_{\mathcal{F}(d,p,s)} \mathbb{E}(L(Y,f(X)))$,

$$\pi_f(z) = \frac{1}{4s+2}\left(L(y,f(x)) - L(y,f_0(x))\right).$$

Define $\Pi = \left\{\pi_f : f \in \mathcal{F}^b(d,p,s)\right\}$, then

$$\sup_{f\in\mathcal{F}^b(d,p,s)} |f| \le (2s+1) \text{ and } \sup_{\pi\in\Pi}|\pi| \le 1.$$

We consider an indexed empirical processes as $P_n\pi - P\pi$. For any $\pi \in \pi$, $P\pi = \mathbb{E}\pi(Z)$, and $P_n\pi = n^{-1}\sum_{i=1}^n \pi(Z_i)$ with Z_i's *i.i.d* from P. We have

$$\begin{aligned}
&\mathbb{P}\left(\text{Risk}\left(\widehat{f}, f^{(d,p)}\right) > (4s+2)4M\right)\\
\le\ &\mathbb{P}\left(\frac{1}{4s+2}\text{Risk}\left(\widehat{f}, f^{(d,p)}\right) > 4M\right)\\
\le\ &\mathbb{P}\left(\sup_{\pi\in\Pi}|P_n\pi - P\pi| > 4M\right)\\
\le\ &\left(2 - \frac{1}{8nM^2}\right)\mathbb{P}\left(\sup_{\pi\in\Pi}\left|\sum_{i=1}^{n}\sigma_i\pi(Z_i)\right| > M\right)(9)
\end{aligned}$$

where σ_i's are i.i.d rademacher variables, independent of Z_i's with $P(\sigma_i = 1) = P(\sigma_i = -1) = \frac{1}{2}$. (9) can be proved by the redemacher symmetrization Bartlett and Mendelson (2002). By conditioning on Z_i's, we can further bound the tail probability by standard chaining trick van der Vaart and Wellner (2000). The chaining trick involves calculating the covering entropy of Π under a $L_2(P_n)$ norm in lemma 5.2. This explains why we get a term $p\log d$ in the numerator of the rate.

Lemma 5.2 *For any $\epsilon > 0$, the ϵ-covering entropy over a function class Π is defined as*

$$\mathcal{N}(\epsilon, \Pi, L_2(P_n)) \le \frac{2p}{\epsilon^2}\log(e + 2e(dp+1)\frac{\epsilon^2}{p}).$$

Proof Consider $\mathcal{G} = \{L(y,f(x)) : f \in \mathcal{F}^b(d,p,s)\}$ and obviously the covering entropy of Π should be upper bounded by that of $\mathcal{G}$. To construct a ϵ-net on $\mathcal{G}$, we first examine the relation ship between $\mathcal{G}$ and $\mathcal{F}^b(d,p,s)$. Since the hinge loss function is Lipschitz continuous with the Lipschitz constant 1, for any Ł and Ł′ $\in \mathcal{G}$, we have

$$\|L - L'\|_P^2 \le \|f - f'\|_P^2,$$

where f and $f' \in \mathcal{F}^b(d,p,s)$. Given a group of basis functions defined as

$$\mathcal{D} = \left\{\xi_{jk+}, \xi_{jk-}, b_+, b_-, j = 1,...,d, k = 1,...,p\right\},$$

where $\xi_{jk+} = \sqrt{p}(2s+1)\psi_{jk}$, $\xi_{jk-} = -\sqrt{p}(2s+1)\psi_{jk}$, $b_+ = \sqrt{p}(2s+1)b$ and $b_- = -\sqrt{p}(2s+1)b$.

For two function sets $\widetilde{\mathcal{F}}$ and $\widetilde{\mathcal{M}}$ spanned by $\mathcal{D}$,

$$\begin{aligned}
\widetilde{\mathcal{F}} \ = \ &\Big\{\widetilde{f} : \widetilde{f} = \sum_{j=1}^{d}\sum_{k=1}^{p}(\lambda_{jk+}\xi_{jk+} + \lambda_{jk-}\xi_{jk-})\\
&+\lambda_{0+}b_+ + \lambda_{0-}b_-, \lambda_{jk+}, \lambda_{jk-} \ge 0,\\
&\sum_{j=1}^{d}\sqrt{\sum_{k=1}^{p}\left(\lambda_{jk+}^2 + \lambda_{jk-}^2\right)} + \lambda_{0+} + \lambda_{0-} \le \frac{1}{\sqrt{p}}\Big\},
\end{aligned}$$

and

$$\begin{aligned}\widetilde{\mathcal{M}} &= \Bigg\{\widetilde{f}:\widetilde{f}=\sum_{j=1}^{d}\sum_{k=1}^{p}(\lambda_{jk+}\xi_{jk+}+\lambda_{jk-}\xi_{jk-})\\ &\quad +\lambda_{0+}b_{+}+\lambda_{0-}b_{-},\lambda_{jk+},\lambda_{jk-}\geq 0,\\ &\quad \sum_{j=1}^{d}\sum_{k=1}^{p}(\lambda_{jk+}+\lambda_{jk-})+\lambda_{0+}+\lambda_{0-}\leq 1\Bigg\},\end{aligned}$$

we can see $\mathcal{F}^b(d,p,s)\subset\widetilde{\mathcal{F}}\subset\widetilde{\mathcal{M}}$. By Lemma 2.6.11 in van der Vaart and Wellner (2000) and transitivity, we have

$$\begin{aligned}&\mathcal{N}\left(\sqrt{p}(4s+2)\epsilon,\mathcal{F}^b(d,p,s),L_2(P)\right)\\ &\leq\mathcal{N}(\sqrt{p}(4s+2)\epsilon,\widetilde{\mathcal{M}},L_2(P))\\ &\leq\frac{2p}{\epsilon^2}\log\left(e+e2(dp+1)\frac{\epsilon^2}{p}\right)\\ &\leq\frac{2p}{\epsilon^2}\log\left(e+e2(d+1)\epsilon^2\right).\end{aligned}$$

Therefore a $(4s+2)\epsilon$-net over $\widetilde{\mathcal{M}}$ induces a $(4s+2)\epsilon$-net in $\mathcal{G}$, which completes the proof.

6 Experimental Results

In this section, we report empirical results on both simulated and real datasets. All the tuning parameters are selected over a grid according to their generalization performance on held-out datasets.

Simulation: We first examine the empirical performance of the SAM in terms of its generalization accuracy and model selection using simulated data sets. We compare the SAM using B-Spline basis against L_1-SVM, the COSSO-SVM using Gaussian kernels and SVM using Gaussian kernels. The generalization error is estimated by Monte Carlo integration using 100,000 test samples from the same distribution as the training samples. We use the following procedure to generate 100 samples:

1. Let $X_j=(W_j+U)/2, j=1,...,d$, where $W_1,...,W_d$ and U are i.i.d. from Uniform$(0,1)$. Therefore the correlation between X_j and X_k is 0.5 for $j\neq k$.

2. We choose two additive function as discriminant functions

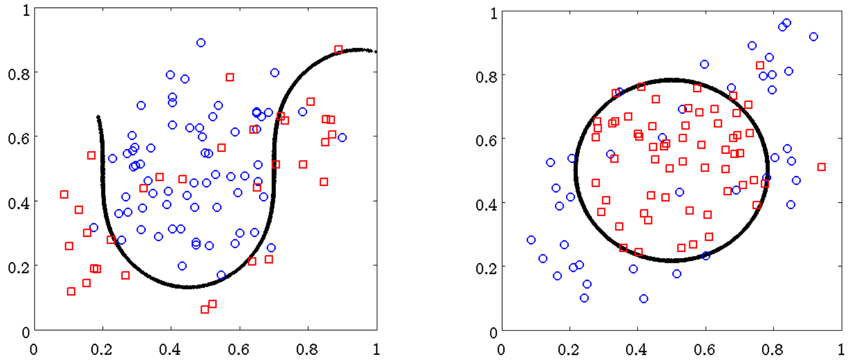
$$\begin{aligned}f(x) &= \sin(2\pi(x_1-0.2))-20(x_2-0.5)^3,\\ f(x) &= (x_1-0.5)^2+(x_2-0.5)^2-0.08.\end{aligned}$$

We assign the label using the discriminant function $Y=\text{sign}(f(X))$ and randomly flip the label with probability 0.1 (Bayes error = 0.1). Figure 3 shows the training data and discriminant functions using the two informative dimensions. We run simulation example 100 times and report the mean and standard errors of the misclassification rates. The performance comparison of the SAM against SVM and L_1-SVM is provided in Table 1.

It can be seen that the SAM and the COSSO have almost the same performance in terms of prediction and variable selection. They both outperform L_1-SVM and SVM under all different settings. As the number of redundant variables increases, the performances of the SAM, the COSSO, and L_1-SVM are shown to be stable, where as SVM deteriorates faster. This is because the SAM is able to remove the redundant features, which in contrast to SVM involving all the variables. This result is consistent with the statistical learning theory. We can also see the limitation of L_1-SVM from the experimental results. Figure 4 and Figure 5 illustrate typical examples of different models using two informative variables when $d=100$. L_1-SVM can only separate data using linear discriminant functions, while the SAM can fit a more flexible discriminant function to better fit the data. Especially when the decision boundary is highly non-linear such as in the second simulation example, L_1-SVM performs much worse than the SAM and even fails to outperform SVM when $d>n$. The SAM shares the advantage of both sparsity and non-linearity and delivers better performance in more complex classification problems.

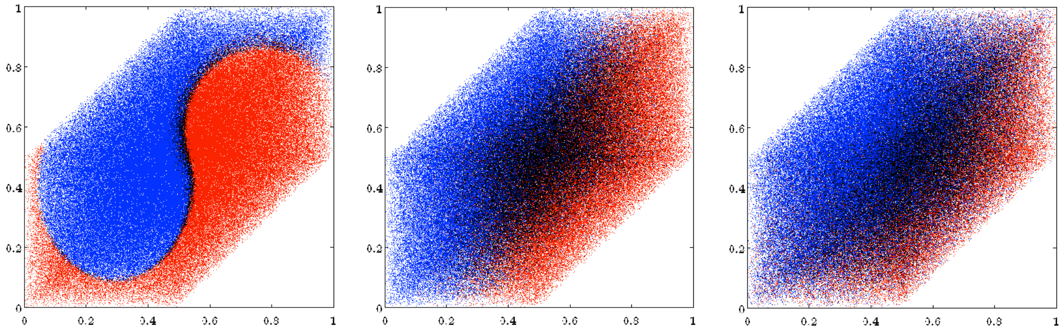
Real Examples: We compare the SAM against the COSSO, L_1-SVM and SVM using three real datasets: (a) the Sonar MR data; (b) the SAM data; and (c) and the Golub data. The Sonar data has has 208 (111:97) samples with 60 variables. We randomly select 140 (75:65) of samples for training and use the remaining 68 (36:32) samples for testing. The Spam dataset has 4601 (1813:2788) samples with 57 variables. We randomly select 300 (150:150) of samples for training and use the remaining 4301 (1663:2638) samples for testing. The original Golub data has 72 (47:25) samples with 7129 variables. As in Dudoit et al. (2002), we preprocessed the Golub data in the following steps: 1) truncation: any expression level was truncated below at 1 and above at 16,000; 2) filtering: any gene was excluded if its max/min $\leq$ 5 and max $-$ min $\leq$ 500, where max and min were the maximum and minimum expression levels of the gene across all samples. Finally, as preliminary gene screening, we selected the top 2000 genes with the largest sample variances across samples. We randomly select 55 (35:20) of samples for training and use the remaining 17 (12:5) samples for testing. Tuning parameters are chosen by 5-fold cross validation on training sets. We do this randomization 30 times and average the testing errors for each model.

As suggested in Table 2, under high-dimensional setting such as the Golub dataset, the SAM still maintain a good performance, and the COSSO failed to obtain



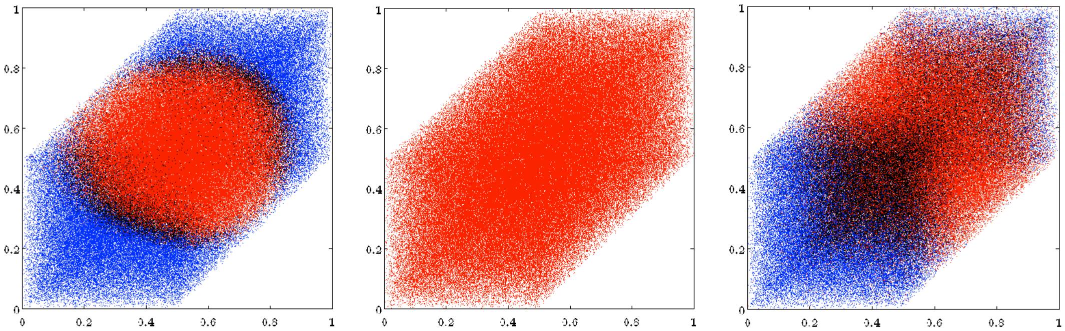
(a) $f(x) = \sin(2\pi(x_1 - 0.2)) - 20(x_2 - 0.5)^3$ (b) $f(x) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.08$

Figure 3: The training data with labels +1 are represented in ◦, while the training data with labels −1 are represented in □. The black curves represent the decision boundaries



(a) SAM (b) L_1-SVM (c) Std SVM

Figure 4: A typical classification result for $f(x) = \sin(2\pi(x_1 - 0.2)) - 20(x_2 - 0.5)^3$ when $d = 100$: blue dots represent the data labeled "+1", red dots represent the data labeled "-1" and black dots represents overlap.



(a) the SAM (b) L_1-SVM (c) Std SVM

Figure 5: A typical classification results for $f(x) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.08$ when $d = 100$: blue dots represent the data classified as "+1", red dots represent the data classified as "-1" and black dots represents overlap.

the results (will be explained later) while the standard SVM completely fails due to a large amount of noise variables. Under low-dimensional setting such as Spam data set, we can see the SAM still outperforms the L_1-SVM and the standard SVM. Although the L_1-SVM tends to yield a sparser solution than the SAM, the better prediction power of the SAM suggest that the L_1-SVM may lose some important variables due to the restriction linear discriminant function. The standard SVM beats the SAM, the COSSO and L_1-SVM on Sonar MR data. One possible explanation is that the sparsity assumption may not hold for this data set. But the SAM still works better than the L_1-SVM on Sonar MR data.

Timing Comparison: We also conduct the timing comparison between the SAM and the COSSO. As

Table 1: Comparison of average testing errors over 100 replications

Models	d	25	50	100	200	400	800
True discriminant function		$f(x) = \sin(2\pi(x_1 - 0.2)) - 20(x_2 - 0.5)^3$					
SAM	Test Error	0.181(0.022)	0.193(0.027)	0.198(0.016)	0.208(0.019)	0.214(0.018)	0.218(0.021)
	# of variables	8.2(2.42)	11.7(3.02)	14.1(4.11)	16.9(5.79)	16.2(5.10)	17.3(7.49)
COSSO	Test Error	0.180(0.029)	0.197(0.031)	0.202(0.017)	0.205(0.018)	0.216(0.022)	0.217(0.020)
	# of variables	9.9(2.88)	14.4(3.65)	15.7(4.27)	17.2(6.11)	17.8(5.51)	18.1(6.51)
L_1-SVM	Test Error	0.304(0.029)	0.313(0.034)	0.301(0.022)	0.306(0.213)	0.324(0.039)	0.323(0.051)
	# of variables	7.75(4.25)	8.25(4.33)	11.2(5.04)	10.5(4.05)	11.7(4.43)	11.8(4.99)
Std SVM	Test Error	0.283(0.013)	0.329(0.018)	0.356(0.024)	0.376(0.013)	0.401(0.018)	0.425(0.025)
	# of variables	25.0(0.00)	50.0(0.00)	100(0.00)	200(0.00)	400(0.00)	400(0.00)
True discriminant function		$f(x) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.08$					
SAM	Test Error	0.180(0.034)	0.201(0.031)	0.207(0.035)	0.214(0.034)	0.232(0.044)	0.233(0.035)
	# of variables	7.1(2.22)	10.1(4.13)	13.4(4.66)	13.3(4.88)	13.3(4.75)	13.8(5.57)
COSSO	Test Error	0.176(0.026)	0.199(0.023)	0.209(0.029)	0.217(0.034)	0.233(0.048)	0.231(0.040)
	# of variables	8.3(2.15)	12.1(3.99)	15.4(5.06)	17.4(6.02)	16.1(5.82)	16.7(6.05)
L_1-SVM	Test Error	0.423(0.006)	0.427(0.017)	0.421(0.007)	0.424(0.021)	0.431(0.025)	0.422(0.031)
	# of variables	2.95(1.73)	2.94(2.07)	3.35(2.03)	4.81(3.86)	5.63(4.69)	6.12(4.55)
Std SVM	Test Error	0.302(0.013)	0.331(0.022)	0.337(0.011)	0.350(0.14)	0.358(0.14)	0.361(0.021)
	# of variables	25.0(0.00)	50.0(0.00)	100(0.00)	200(0.00)	400(0.00)	400(0.00)

Table 2: Comparison of 5-fold double cross validation errors

Data	Sonar MR		Spam		Golub	
Models	Test Error	# of variables	Test Error	# of variables	Test Error	# of variables
the SAM	0.191(0.051)	48.4(6.54)	**0.905(0.008)**	34.3(5.78)	**0.018(0.029)**	40.1(7.25)
the COSSO	0.189(0.055)	55.4(7.73)	0.922(0.010)	36.3(6.11)	N.A.	N.A.
L_1-SVM	0.266(0.044)	24.1(14.5)	0.132(0.018)	38.2(6.30)	0.053(0.047)	35.2(4.33)
Std SVM	**0.135(0.031)**	60.0(0.00)	0.155(0.024)	57.0(0.00)	0.294(0.000)	2000(0.00)

Table 3: Timing comparison

Data	Sonar MR		Spam		Golub	
Models	Timing	# of parameters	Timing	# of parameters	Timing	# of parameters
SAM	55.10(7.21)	181	72.44(9.77)	172	1415(66.4)	6001
COSSO	2180(141)	8401	5412(181)	17101	N.A.	110001

there is no package available for the COSSO, we also apply the accelerated proximal gradient descent algorithm to the COSSO. All codes use the same setting: double precision with a convergence threshold 1e-3. We choose the difference of empirical means between two classes as the bandwidth parameter in Gaussian kernel for the COSSO and $p = n^{1/5}$ for the SAM. The range of regularization parameters is chosen so that each method produced approximately the same number of non-zero estimates.

The timing results can be seen in Table 3 showing that the SAM outperforms the COSSO in timing for all 3 datasets. Since we adopt the truncation rate as $p_n = \mathcal{O}(n^{1/5})$, it allows p_n to increase very slowly as the sample size increases. On the contrast, the COSSO requires nd parameters (linearly increasing in n), which cannot scale up to larger problems, especially when n is relatively large. Therefore it is not surprising to see the SAM is much more scalable than the COSSO in practice. In our experiments, the spam data set has the largest training sample size among all three datasets, 300 yielding $p \approx 3$ and 172 parameters in total, but the COSSO requires almost 100 times more parameters than the SAM. For the Golub data set, the minimization problem of the COSSO involving 110001 parameters (including the intercept), which is about 16 times larger than that of the SAM. Eventually. the timing of the COSSO exceeds our time limit 18000 seconds (5 hours) and fails to get the results.

7 Conclusions

This article proposes the sparse additive machine that simultaneously perform nonparametric variable selection and classification. In particular, the method, together with the computational algorithms, provides another recipe for high dimensional, small sample size and complex data analysis, that can be difficult for conventional methods. The proposed method has been shown to perform well as long as p does not grow too fast and the discriminant function has a sparse representation. In many problems, our method significantly outperforms standard SVM and L_1-SVM and is much more scalable than the COSSO.

References

Bach, F. (2008). Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research* **9** 1179–1225.

Bartlett, P. and Mendelson, S. (2002). Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research* **3** 463–482.

Boyd, S. and Vandenberghe, L. (2009). *Convex Optimization.* 2nd ed. Cambridge University.

Bradley, P. and Mangasarian, O. (1998). Feature selection via concave minimization and support vector machines. *International Conference on Machine Learning* 82–90.

Christmann, A. and Hable, R. . (2010). Support vector machines for additive models: Consistency and robustness. *Manuscript, http://arxiv.org/abs/1007.406* .

Dudoit, S., Fridly, J. and Speed, T. (2002). Comparison of discrimination methods for the classification of tumors using expression data. *Journal of the American Statistical Association* **97** 77–87.

Greenshtein, E. and Ritov, Y. (2004). Persistency in high dimensional linear predictor-selection and the virtue of over-parametrization. *Journal of Bernoulli* **10** 971–988.

Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning* **46** 389–422.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements of Statistical Learning Data Mining, Inference, and Prediction.* 2nd ed. Springer-Verlag.

Kohavi, R. and John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence* 273–324.

Koltchinskii, V. and Yuan, M. (2010). Sparsity in multiple kernel learning. *Annals of Statistics* **38** 3660–3695.

Lin, Y. and Zhang, H. (2006). Component selection and smoothing in multivariate nonparametric regression. *Annals of Statistics* **34** 2272–2297.

Liu, H., Lafferty, J. and Wasserman, L. (2008). Nonparametric regression and classification with joint sparsity constraints. *Advances in Neural Information Processing Systems* 969–976.

Liu, H. and Zhang, J. (2009). On the estimation consistency of the group lasso and its applications. *International Conference on Artificial Intelligence and Statistics* **5** 376–383.

Meier, L., Geer, S. V. D. and Bühlmann, P. (2009). High-dimensional additive modeling. *Annals of Statistics* **37** 3779–3821.

Nesterov, Y. (2005). Smooth minimization of non-smooth functions. mathematical programming. *Mathematical Programming* **103** 127–152.

Ravikumar, P., Lafferty, J., Liu, H. and Wasserman, L. (2009). Sparse additive models. *Journal of the Royal Statistical Society, Series B* **71** 1009–1030.

van der Vaart, A. and Wellner, J. (2000). *Weak Convergence and Empirical Processes with Application to Statistics.* 2nd ed. Springer-Verlag.

Vapnik, V. (1998). *Statistical Learning Theory.* Wiley-Interscience.

Wahba, G. (1999). Support vector machines, reproducing kernel hilbert spaces and the randomized gacv. *Advances in Kernel Methods: Support Vector Learning* 69–88.

Wang, L. and Shen, X. (2007). On l_1-norm multiclass support vector machines: Methodology and theory. *Journal of the American Statistical Association* **102** 583–594.

Zhang, H. (2006). Variable selection for support vector machines via smoothing spline anova. *Statistica Sinica* **16** 659–674.

Zhu, J., Hastie, T., Rosset, S. and Tibshirani, R. (2003). 1-norm support vector machines. *Advances in Neural Information Processing Systems* .