Additive Approximations in High Dimensional Nonparametric Regression via the SALSA

Kirthevasan Kandasamy KANDASAMY@CS.CMU.EDU
Yaoliang Yu YAOLIANG@CS.CMU.EDU
Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

High dimensional nonparametric regression is an inherently difficult problem with known lower bounds depending exponentially in dimension. A popular strategy to alleviate this curse of dimensionality has been to use additive models of *first order*, which model the regression function as a sum of independent functions on each dimension. Though useful in controlling the variance of the estimate, such models are often too restrictive in practical settings. Between non-additive models which often have large variance and first order additive models which have large bias, there has been little work to exploit the trade-off in the middle via additive models of intermediate order. In this work, we propose SALSA, which bridges this gap by allowing interactions between variables, but controls model capacity by limiting the order of interactions. SALSA minimises the residual sum of squares with squared RKHS norm penalties. Algorithmically, it can be viewed as Kernel Ridge Regression with an additive kernel. When the regression function is additive, the excess risk is only polynomial in dimension. Using the Girard-Newton formulae, we efficiently sum over a combinatorial number of terms in the additive expansion. Via a comparison on 15 real datasets, we show that our method is competitive against 21 other alternatives.

1. Introduction

Given i.i.d samples $(X_i, Y_i)_{i=1}^n$ from some distribution $\mathbb{P}_{XY}$, on $\mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^D \times \mathbb{R}$, the goal of least squares regression is to estimate the regression function $f_*(x) = \mathbb{E}[Y|X = x]$. A popular approach is linear regression which models f_* as a linear combination of the variables x, i.e. $f(x) = \beta^\top x$ for some $\beta \in \mathbb{R}^D$. Linear Regression is typically solved by minimising the sum of squared errors on the training set subject to a complexity penalty on β. Such *parametric* methods are conceptually simple and have desirable statistical properties when the problem meets the assumption. However, the parametric assumption is generally too restrictive for many real problems.

Nonparametric regression refers to a suite of methods that typically only assume smoothness on f_*. They present a more compelling framework for regression since they encompass a richer class of functions than parametric models do. However they suffer from severe drawbacks in high dimensional settings. The excess risk of nonparametric methods has exponential dependence on dimension. Current lower bounds (Györfi et al., 2002) suggest that this dependence is unavoidable. Therefore, to make progress stronger assumptions on f_* beyond just smoothness are necessary. In this light, a common simplification has been to assume that f_* decomposes into the additive form $f_*(x) = f_*^{(1)}(x_1) + f_*^{(2)}(x_2) + \cdots + f_*^{(D)}(x_D)$ (Hastie & Tibshirani, 1990; Lafferty & Wasserman, 2005; Ravikumar et al., 2009). In this exposition, we refer to such models as *first order* additive models. Under this assumption, the excess risk improves significantly.

That said, the first order assumption is often too biased in practice since it ignores interactions between variables. It is natural to ask if we could consider additive models which permit interactions. For instance, a second order model has the expansion $f_*(x) = f_*^{(1)}(x_1, x_2) + f_*^{(2)}(x_1, x_3) + \dots$. In general, we may consider d orders of interaction which have $\binom{D}{d}$ terms in the expansion. If $d \ll D$, we may allow for a richer class of functions than first order models, and hopefully still be able to control the excess risk.

Even when f_* is not additive, using an additive approximation has its advantages. It is a well understood statistical concept that when we only have few samples, using a simpler model to fit our data gives us a better trade-off for variance against bias. Since additive models are *statistically simpler* they may give us better estimates due to reduced

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

variance. In most nonparametric regression methods, the bias-variance trade-off is managed via a parameter such as the bandwidth of a kernel or a complexity penalty. In this work, we demonstrate that this trade-off can also be controlled via additive models with different orders of interaction. Intuitively, we might use low order interactions with few data points but with more data we can increase model capacity via higher order interactions. Indeed, our experiments substantiate this intuition: additive models do well on several datasets in which f_* is not necessarily additive.

There are **two key messages in this paper**. The first is that we should use additive models in high dimensional regression to reduce the variance of the estimate. The second is that it is necessary to model beyond just first order models to reduce the bias. Our contributions in this paper are:

1. We formulate additive models for nonparametric regression beyond first order models. Our method SALSA – for *Shrunk Additive Least Squares Approximation*– estimates a d^{th} order additive function containing $\binom{D}{d}$ terms in its expansion. Despite this, the computational complexity of SALSA is $\mathcal{O}(Dd^2)$.
2. Our theoretical analysis bounds the excess risk for SALSA for (i) additive f_* under reproducing kernel Hilbert space assumptions and (ii) non-additive f_* in the agnostic setting. In (i), the excess risk has only polynomial dependence on D.
3. We compare our method against 21 alternatives on synthetic and 15 real datasets. SALSA is more consistent and in many cases outperforms other methods. Our software and datasets are available at `github.com/kirthevasank/salsa`. Our implementation of locally polynomial regression is also released as part of this paper and is made available at `github.com/kirthevasank/local-poly-reg`.

Before we proceed we make an essential observation. When parametric assumptions are true, parametric regression methods can scale both statistically and computationally to possibly several thousands of dimensions. However, it is common knowledge in the statistics community that nonparametric regression can be reliably applied only in very low dimensions with reasonable data set sizes. Even $D = 10$ is considered "high" for nonparametric methods. In this work we aim to statistically scale nonparametric regression to dimensions on the order 10–100 while addressing the computational challenges in doing so.

Related Work

A plurality of work in high dimensional regression focuses on first order additive models. One of the most popular techniques is the back-fitting algorithm (Hastie et al., 2001) which iteratively approximates f_* via a sum of D one dimensional functions. Some variants such as RODEO (Lafferty & Wasserman, 2005) and SpAM (Ravikumar et al., 2009) study first order models in variable selection/sparsity settings. MARS (Friedman, 1991) uses a sum of splines on individual dimensions but allows interactions between variables via products of hinge functions at selected knot points. Lou et al. (2013) model f_* as a first order model plus a sparse collection of pairwise interactions. However, restricting ourselves to only to a sparse collection of second order interactions might be too biased in practice. COSSO (Lin & Zhang, 2006) study higher order models but when you need only a sparse collection of them. In Section 4 we list several other parametric and nonparametric methods used in regression.

Our approach is based on additive kernels and builds on Kernel Ridge Regression (Steinwart & Christmann, 2008; Zhang, 2005). Using additive kernels to encode and identify structure in the problem is fairly common in Machine Learning literature. A large line of work, in what has to come to be known as Multiple Kernel Learning (MKL), focuses on precisely this problem (Bach, 2008; Gönen & Alpaydin, 2011; Xu et al., 2010). Additive models have also been studied in Gaussian process literature via additive kernels (Duvenaud et al., 2011; Plate, 1999). However, they treat the additive model just as a heuristic whereas we also provide a theoretical analysis of our methods.

2. Preliminaries

We begin with a brief review of some background material. We are given i.i.d data $(X_i, Y_i)_{i=1}^n$ sampled from some distribution $\mathbb{P}_{XY}$ on a compact space $\mathcal{X} \times \mathcal{Y} \subset \mathbb{R}^D \times \mathbb{R}$. Let the marginal distribution of X on $\mathcal{X}$ be $\mathbb{P}_X$ and the $L_2(\mathbb{P}_X)$ norm be $\|f\|_2^2 = \int f^2 \mathrm{d}\mathbb{P}_X$. We wish to use the data to find a function $f : \mathcal{X} \to \mathbb{R}$ with small risk

$$\mathcal{R}(f) = \int_{\mathcal{X}\times\mathcal{Y}} (y - f(x))^2 \mathrm{d}\mathbb{P}_{XY}(x,y) = \mathbb{E}[(Y - f(X))^2].$$

It is well known that $\mathcal{R}$ is minimised by the regression function $f_*(\cdot) = \mathbb{E}_{XY}[Y|X = \cdot]$ and the *excess risk* for any f is $\mathcal{R}(f) - R(f_*) = \|f - f_*\|_2^2$ (Györfi et al., 2002). Our goal is to develop an estimate that has low expected excess risk $\mathbb{E}\mathcal{R}(\hat{f}) - \mathcal{R}(f_*) = \mathbb{E}[\|\hat{f} - f_*\|_2^2]$, where the expectation is taken with respect to realisations of the data $(X_i, Y_i)_{i=1}^n$.

Some smoothness conditions on f_* are required to make regression tractable. A common assumption is that f_* has bounded norm in the reproducing kernel Hilbert space (RKHS) $\mathcal{H}_\kappa$ of a continuous positive definite kernel $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. By Mercer's theorem (Schölkopf & Smola, 2001), κ permits an eigenexpansion of the form $\kappa(x, x') = \sum_{j=1}^\infty \mu_j \phi_j(x)\phi_j(x')$ where $\mu_1 \geq \mu_2 \geq \cdots \geq 0$ are the eigenvalues of the expansion and $\phi_1, \phi_2, \ldots$ are an orthonormal basis for $L^2(\mathbb{P}_X)$.

Kernel Ridge Regression (KRR) is a popular technique for nonparametric regression. It is characterised as the solution of the following optimisation problem over the RKHS of some kernel κ.

$$\hat{f} = \operatorname*{argmin}_{f\in\mathcal{H}_\kappa} \lambda\|f\|^2_{\mathcal{H}_\kappa} + \frac{1}{n}\sum_{i=1}^{n}(Y_i - f(X_i))^2. \tag{1}$$

Here λ is the regularisation coefficient to control the variance of the estimate and is decreasing with more data. Via the representer theorem (Schölkopf & Smola, 2001; Steinwart & Christmann, 2008), we know that the solution lies in the linear span of the canonical maps of the training points X_1^n – i.e. $\hat{f}(\cdot) = \sum_i \alpha_i \kappa(\cdot, X_i)$. This reduces the above objective to $\hat{\alpha} = \operatorname{argmin}_{\alpha\in\mathbb{R}^n} \lambda\alpha^\top K\alpha + \frac{1}{n}\|Y - K\alpha\|_2^2$ where $K \in \mathbb{R}^{n\times n}$ is the kernel matrix with $K_{ij} = \kappa(X_i, X_j)$. The problem has the closed form solution $\hat{\alpha} = (K + \lambda nI)^{-1}Y$. KRR has been analysed extensively under different assumptions on f_*; see (Steinwart & Christmann, 2008; Steinwart et al., 2009; Zhang, 2005) and references therein. Unfortunately, as is the case with many nonparametric methods, KRR suffers from the curse of dimensionality as its excess risk is exponential in D.

Additive assumption: To make progress in high dimensions, we assume that f_* decomposes into the following additive form that contains interactions of d orders among the variables. (Later on, we will analyse non-additive f_*.)

$$f_*(x) = \sum_{1\le i_1<i_2<\cdots<i_d\le D} f_*^{(j)}(x_{i_1}, x_{i_2}, \ldots, x_{i_d}), \tag{2}$$

We will write, $f_*(x) = \sum_{j=1}^{M_d} f_*^{(j)}(x^{(j)})$ where $M_d = \binom{D}{d}$, and $x^{(j)}$ denotes the subset $(x_{i_1}, x_{i_2}, \ldots, x_{i_d})$. We are primarily interested in the setting $d \ll D$. While there are a large number of $f_*^{(j)}$'s, each of them only permits interactions of at most d variables. We will show that this assumption does in fact reduce the statistical complexity of the function to be estimated. The first order additive assumption is equivalent to setting $d = 1$ above. A potential difficulty with the above assumption is the combinatorial computational cost in estimating all $f_*^{(j)}$'s when $d > 1$. We circumvent this bottleneck using two strategems: a classical result from RKHS theory, and a computational trick using elementary symmetric polynomials used before by Duvenaud et al. (2011); Shawe-Taylor & Cristianini (2004) in the kernel literature for additive kernels.

3. SALSA

To extend KRR to additive models we first define kernels $k^{(j)}$ that act on each subset $x^{(j)}$. We then optimise the following objective jointly over $\hat{f}^{(j)} \in \mathcal{H}_{k^{(j)}}, j = 1\ldots, M_d$.

$$\{\hat{f}^{(j)}\}_{j=1}^{M_d} = \operatorname*{argmin}_{f^{(j)}\in\mathcal{H}_{k^{(j)}}, j=1,\ldots,M_d} \lambda\sum_{j=1}^{M_d}\|f^{(j)}\|^2_{\mathcal{H}_{k^{(j)}}} + \frac{1}{n}\sum_{i=1}^{n}\Big(Y_i - \sum_{j=1}^{M_d} f^{(j)}(X_i^{(j)})\Big)^2. \tag{3}$$

Our estimate for f is then $\hat{f}(\cdot) = \sum_j \hat{f}^{(j)}(\cdot)$. At first, this appears troublesome since it requres optimising over nM_d parameters $(\alpha_i^{(j)}), j = 1, \ldots, M_d, i = 1, \ldots, n$. However, from the work of Aronszajn (1950), we know that the solution of (3) lies in the RKHS of the sum kernel k

$$\begin{aligned} k(x, x') &= \sum_{j=1}^{M_d} k^{(j)}(x^{(j)}, x^{(j)\prime}) \qquad (4)\\ &= \sum_{1\le i_1<\cdots<i_d\le D} k^{(j)}([x_{i_1}, \ldots, x_{i_d}], [x'_{i_1}, \ldots, x'_{i_d}]). \end{aligned}$$

See Remark 6 in Appendix A for a proof. Hence, the solution $\hat{f}$ can be written in the form $\hat{f}(\cdot) = \sum_i \alpha_i k(\cdot, X_i)$ This is convenient since we only need to optimise over n parameters despite the combinatorial number of kernels. Moreover, it is straightforward to see that the solution is obtained by solving (1) by plugging in the sum kernel k for κ. Consequently $\hat{f}^{(j)} = \sum_i \hat{\alpha}_i k^{(j)}(\cdot, X_i^{(j)})$ and $\hat{f} = \sum_i \hat{\alpha}_i k(\cdot, X_i)$ where $\hat{\alpha}$ is the solution of (1). While at first sight the differences with KRR might seem superficial, we will see that the *stronger* additive assumption will help us reduce the excess risk for high dimensional regression. Our theoretical results will be characterised directly via the optimisation objective (3).

3.1. The ESP Kernel

While the above formulation reduces the number of optimisation parameters, the kernel still has a combinatorial number of terms which can be expensive to compute. While this is true for arbitrary choices for $k^{(j)}$'s, under some restrictions we can efficiently compute k. For this, we use the same trick used by Shawe-Taylor & Cristianini (2004) and Duvenaud et al. (2011). First consider a set of base kernels acting on each dimension $k_1, k_2, \ldots, \ldots, k_D$. Define $k^{(j)}$ to be the product kernel of all kernels acting on each coordinate – $k^{(j)}(x^{(j)}, x^{(j)\prime}) = k_{i_1}(x_{i_1}, x'_{i_1})k_{i_2}(x_{i_2}, x'_{i_2})\cdots k_{i_d}(x_{i_d}, x'_{i_d})$. Then, the additive kernel $k(x, x')$ becomes the d^{th} elementary symmetric polynomial (ESP) of the D variables $k_1(x_1, x'_1), \ldots, k_D(x_D, x'_D)$. Concretely,

$$k(x, x') = \sum_{1\le i_1<i_2<\cdots<i_d\le D}\left(\prod_{\ell=1}^{d} k_{i_\ell}(x_{i_\ell}, x'_{i_\ell})\right). \tag{5}$$

We refer to (5) as the ESP kernel. Using the Girard-Newton identities (Macdonald, 1995) for ESPs, we can compute this summation efficiently. For the D variables $s_1^D = s_1, \ldots, s_D$ and $1 \leq m \leq D$, define the m^{th} power sum p_m and the m^{th} elementary symmetric polynomial e_m:

$$p_m(s_1^D) = \sum_{i=1}^{D} s_i^m \ ,$$
$$e_m(s_1^D) = \sum_{1\leq i_1<i_2<\cdots<i_m\leq D} s_{i_1} \times s_{i_2} \times \cdots \times s_{i_m}.$$

In addition define $e_0(s_1^n) = 1$. Then, the Girard-Newton formulae state,

$$e_m(s_1^D) = \frac{1}{m}\sum_{i=1}^{m}(-1)^{i-1}e_{m-i}(s_1^D)p_i(s_1^D).$$

Starting with $m = 1$ and proceeding up to $m = d$, e_d can be computed iteratively in just $\mathcal{O}(Dd^2)$ time. By treating $s_i = k_i$, the kernel matrix can be computed in $\mathcal{O}(n^2d^2D)$ time. While the ESP trick restricts the class of kernels we can use in SALSA, it applies for important kernel choices. For example, if each $k^{(j)}$ is a Gaussian kernel, then it is an ESP kernel if we set the bandwidths appropriately.

In what follows, we refer to a kernel such as k (5) which permits only d orders of interaction as a d^{th} order kernel. A kernel which permits interactions of all D variables is of D^{th} order. Note that unlike in MKL, here we do not wish to *learn* the kernel. We use additive kernels to explicitly reduce the complexity of the function class over which we optimise for $\hat{f}$. Next, we present our theoretical results.

3.2. Theoretical Analysis

We first consider the setting when $f_*^{(j)}$ is in $\mathcal{H}_{k^{(j)}}$ over which we optimise for $\hat{f}^{(j)}$. Theorem 3 generally bounds the excess risk of $\hat{f}$ (3) in terms of RKHS parameters. Then, we specialise it to specific RKHSs in Theorem 4 and show that in many cases, the dependence on D reduces from exponential to polynomial for additive f_*. We begin with some assumptions.

Assumption 1. *f_* has a decomposition $f_*(x) = \sum_{j=1}^{M_d} g^{(j)}(x^{(j)})$ where each $g^{(j)} \in \mathcal{H}_{k^{(j)}}$.*

We point out that the decomposition $\{g^{(j)}\}$ need not be unique. To enforce definiteness (by abusing notation) we define $f_*^{(j)} \in \mathcal{H}_{k^{(j)}}, j = 1, \ldots, M_d$ to be the set of functions which minimise $\sum_j \|g^{(j)}\|^2_{\mathcal{H}_{k^{(j)}}}$. Denote the minimum value by $\|\mathbf{f}_*\|^2_{\mathcal{F}}$. We denote it by a norm for reasons made clear in our proofs.

Let $k^{(j)}$ have an eigenexpansion $k^{(j)}(x^{(j)}, x^{(j)\prime}) = \sum_{\ell=1}^{\infty} \mu_\ell^{(j)} \phi_\ell^{(j)}(x^{(j)})\phi_\ell^{(j)}(x^{(j)\prime})$ in $L^2(\mathbb{P}_{X^{(j)}})$. Here, $\{(\phi_\ell^{(j)})_{\ell=1}^{\infty}\}$ is an orthonormal basis for $L^2(\mathbb{P}_{X^{(j)}})$ and $\{(\mu_\ell^{(j)})_{\ell=1}^{\infty}\}$ are its eigenvalues. $\mathbb{P}_{X^{(j)}}$ is the marginal distribution of the coordinates $X^{(j)}$. We also need the following regularity condition on the tail behaviour of the basis functions $\{\phi_\ell^{(j)}\}$ for all $k^{(j)}$. Similar assumptions are made in (Zhang et al., 2013) and are satisfied for a large range of kernels including those in Theorem 4.

Assumption 2. *For some $q \geq 2$, $\exists\ \rho < \infty$ such that for all $j = 1, \ldots, M_d$ and $\ell \in \mathbb{N}$, $\mathbb{E}[\phi_\ell^{(j)}(X)^{2q}] \leq \rho^{2q}$.*

We also define the following,

$$\gamma^{(j)}(\lambda) = \sum_{\ell=1}^{\infty} \frac{1}{1 + \lambda/\mu_\ell^{(j)}}, \quad \gamma_k(\lambda) = \sum_{j=1}^{M_d} \gamma^{(j)}(\lambda). \quad (6)$$

The first term is known as the effective data dimensionality of $k^{(j)}$ (Zhang, 2005; Zhang et al., 2013) and captures the statistical difficulty of estimating a function in $\mathcal{H}_{k^{(j)}}$. γ_k is the sum of the $\gamma^{(j)}$'s. Our first theorem below bounds the excess risk of $\hat{f}$ in terms $\|\mathbf{f}_*\|^2_{\mathcal{F}}$ and γ_k.

Theorem 3. *Let Assumptions 1 and 2 hold. and Y have bounded conditional variance: $\mathbb{E}[(Y - f_*(X))^2|X] \leq \sigma^2$. Then the solution $\hat{f}$ of (3) satisfies,*

$$\mathbb{E}[\mathcal{R}(\hat{f})] - \mathcal{R}(f_*) \leq M_d\left(20\lambda\|\mathbf{f}_*\|^2_{\mathcal{F}} + \frac{12\sigma^2\gamma_k(\lambda)}{n} + \chi(k)\right).$$

Here $\chi(k)$ are kernel dependent low order terms and are given in (11) in Appendix A. Our proof technique generalises the analysis of Zhang et al. (2013) for KRR to the additive case. We use ideas from Aronszajn (1950) to handle sum RKHSs. We consider a space $\mathcal{F}$ containing the tuple of functions $f^{(j)} \in \mathcal{H}_{k^{(j)}}$ and use first order optimality conditions of (3) in $\mathcal{F}$. The proof is given in Appendix A.

The term $\gamma_k(\lambda)$, which typically has exponential dependence on d, arises through the variance calculation. Therefore, by using small d we may reduce the variance of our estimate. However, this will also mean that we are only considering a smaller function class and hence suffer large bias *if* f_* is not additive. In naive KRR, using a D^{th} order kernel (equivalent to setting $M_d = M_D = 1$) the excess risk depends exponentially in D. In contrast, for an additive d^{th} order kernel, $\gamma_k(\lambda)$ has polynomial dependence on D *if* f_* is additive. We make this concrete via the following theorem.

Theorem 4. *Assume the same conditions as Theorem 3. Then, suppressing* $\log(n)$ *terms,*

- *if each $k^{(j)}$ has eigendecay $\mu_\ell^{(j)} \in \mathcal{O}(\ell^{-2s/d})$, then by choosing $\lambda \asymp n^{\frac{-2s}{2s+d}}$, we have $\mathbb{E}[\mathcal{R}(\hat{f})] - \mathcal{R}(f_*) \in \mathcal{O}(D^{2d}n^{\frac{-2s}{2s+d}})$,*

- *if each $k^{(j)}$ has eigendecay $\mu_\ell^{(j)} \in \mathcal{O}(\tilde{\pi}^d \exp(-\alpha\ell^2))$ for some constants $\tilde{\pi}, \alpha$, then by choosing $\lambda \asymp 1/n$, we have $\mathbb{E}[\mathcal{R}(\hat{f})] - \mathcal{R}(f_*) \in \mathcal{O}(\frac{D^{2d}\tilde{\pi}^d}{n})$.*

We bound γ_k via bounds for $\gamma^{(j)}$ and use it to derive the optimal rates for the problem. The proof is in Appendix B.

It is instructive to compare the rates for the cases above when we use a D^{th} order kernel κ in KRR to estimate a non-additive function. The first eigendecay is obtained if each $k^{(j)}$ is a Matérn kernel. Then $f^{(j)}$ belongs to the Sobolev class of smoothness s (Berlinet & Thomas-Agnan, 2004; Tsybakov, 2008). By following a similar analysis, we can show that if κ is in a Sobolev class, then the excess risk of KRR is $\mathcal{O}(n^{\frac{-2s}{2s+D}})$ which is significantly slower than ours. In our setting, the rates are only exponential in d but we have an additional D^{2d} term as we need to estimate several such functions. An example of the second eigendecay is the Gaussian kernel with $\tilde{\pi} = \sqrt{2\pi}$ (Williamson et al., 2001). In the nonadditve case, the excess risk is in the Gaussian RKHS is $\mathcal{O}\big(\frac{(2\pi)^{D/2}}{n}\big)$ which is slower than SALSA whose dependence on D is just polynomial. D, d do not appear in the exponent of n because the Gaussian RKHS contains very smooth functions. KRR is slower since we are optimising over the very large class of non-additive functions and consequently it is a difficult statistical problem. The faster rates for SALSA should not be surprising since the class of additive functions is smaller. The advantage of SALSA is its ability to recover the function at a faster rate when f_* is additive. Finally we note that by taking each base kernel k_i in the ESP kernel to be a 1D Gaussian, each $k^{(j)}$ is a Gaussian. However, at this point it is not clear to us if it is possible to recover a s-smooth Sobolev class via the tensor product of s-smooth one dimensional kernels.

Finally, we analyse SALSA under more agnostic assumptions. We will neither assume that f_* is additive nor that it lies in any RKHS. First, define the functions $f_\lambda^{(j)}$, $j = 1, \dots, M$ which minimise the population objective.

$$\{f_\lambda^{(j)}\}_{j=1}^{M_d} = \underset{f^{(j)} \in \mathcal{H}_{k^{(j)}}, j=1,\dots,M}{\operatorname{argmin}} \lambda \sum_{j=1}^{M_d} \|f^{(j)}\|^2_{\mathcal{H}_{k^{(j)}}} + \mathbb{E}\left[\left(Y - \sum_{j=1}^{M_d} f^{(j)}(X^{(j)})\right)^2\right]. \quad (7)$$

Let $f_\lambda = \sum_j f_\lambda^{(j)}$, $R_\lambda^{(j)} = \|f_\lambda^{(j)}\|_{\mathcal{H}_{k^{(j)}}}$ and $R^2_{d,\lambda} = \sum_j {R_\lambda^{(j)}}^2$. To bound the excess risk in the agnostic setting we also define the class,

$$\mathcal{H}_{d,\lambda} = \{f : \mathcal{X} \to \mathbb{R};\ f(x) = \sum_j f^{(j)}(x^{(j)}), \quad (8)$$
$$\forall j,\ f^{(j)} \in \mathcal{H}_{k^{(j)}}, \|f^{(j)}\|_{\mathcal{H}_{k^{(j)}}} \le R_\lambda^{(j)}\}.$$

Theorem 5. *Let f_* be an arbitrary measurable function and Y have bounded fourth moment $\mathbb{E}[Y^4] \le \nu^4$. Further each $k^{(j)}$ satisfies Assumption 2. Then $\forall\ \eta > 0$,*

$$\mathbb{E}[\mathcal{R}(\hat{f})] - \mathcal{R}(f_*) \le (1+\eta)\mathbf{AE}\ +\ (1+1/\eta)\mathbf{EE},$$

$$\text{where, } \mathbf{AE} = \inf_{f \in \mathcal{H}_{d,\lambda}} \|f - f_*\|_2^2, \quad \mathbf{EE} \in \mathcal{O}\Big(\frac{M_d \gamma_k(\lambda)}{n}\Big).$$

The proof, given in Appendix C, also follows the template in Zhang et al. (2013). Loosely, we may interpret **AE** and **EE** as the approximation and estimation errors[1]. We may use Theorem 5 to understand the trade-offs in approximaing a non-additive function via an additive model. We provide an intuitive "not-very-rigorous" explanation. $\mathcal{H}_{d,\lambda}$ is typically increasing with d since higher order additive functions contain lower order functions. Hence, **AE** is decreasing with d as the infimum is taken over a larger set. On the other hand, **EE** is increasing with d. With more data **EE** decreases due to the $1/n$ term. Hence, we can afford to use larger d to reduce **AE** and balance with **EE**. This results in an overall reduction in the excess risk.

The actual analysis would be more complicated since $\mathcal{H}_{d,\lambda}$ is a bounded class depending intricately on λ. It also depends on the kernels $k^{(j)}$, which differ with d. To make the above intuition concrete and more interpretable, it is necessary to have a good handle on **AE**. However, if we are to overcome the exponential dependence in dimension, usual nonparametric assumptions such as Hölderian/ Sobolev conditions alone will not suffice. Current lower bounds suggest that the exponential dependence is unavoidable (Györfi et al., 2002; Tsybakov, 2008). Additional assumptions will be necessary to demonstrate faster convergence. Once we control **AE**, the optimal rates can be obtained by optimising the bound over η, λ. We wish to pursue this in future work.

3.3. Practical Considerations

Choice of Kernels: The development of our algorithm and our analysis assume that the k_i's are known. This is hardly the case in reality and they have to be chosen properly for good empirical performance. Cross validation is not feasible here as there are too many hyper-parameters. In our experiments we set each k_i to be a Gaussian kernel $k_i(x_i, x_i') = \sigma_Y \exp(-(x_i - x_i')^2/2h_i^2)$ with bandwidth $h_i = c\sigma_i n^{-1/5}$. Here σ_i is the standard deviation of the i^{th} covariate and σ_Y is the standard deviation of Y. The choice of bandwidth was inspired by several other kernel methods which use bandwidths on the order $\sigma_i n^{-1/5}$ (Ravikumar et al., 2009; Tsybakov, 2008). The constant c was hand tuned – we found that performance was robust to choices between 5 and 60. In our experiments we use $c = 20$. c

[1] Loosely (and not strictly) since $\hat{f}$ need not be in $\mathcal{H}_{d,\lambda}$.

was chosen by experimenting on a collection of synthetic datasets and then used in all our experiments. Both synthetic and real datasets used in experiments are independent of the data used to tune c.

Choice of d, λ: If the additive order of f_* is known and we have sufficient data then we can use that for d in (5). However, this is usually not the case in practice. Further, even in non-additive settings, we may wish to use an additive model to improve the variance of our estimate. In these instances, our approach to choose d uses cross validation. For a given d we solve (1) for different λ and pick the best one via cross validation. To choose the optimal d we cross validate on d. In our experiments we observed that the cross validation error had bi-monotone like behaviour with a unique local optimum on d. Since the optimal d was typically small we search by starting at $d = 1$ and keep increasing until the error begins to increase again. If d could be large and linear search becomes too expensive, a binary search like procedure on $\{1, \dots, D\}$ can be used.

We conclude this section with a couple of remarks. First, we could have considered an alternative additive model which sums all interactions up to d^{th} order instead of just the d^{th} order. The excess risk of this model differs from Theorems 3, 4 and 5 only in subdominant terms and/or constant factors. The kernel can be computed efficiently using the same trick by summing all polynomials up to d. In our experiments we found that both our original model (2) and summing over all interactions performed equally well. For simplicity, results are presented only for the former.

Secondly, as is the case with most kernel methods, SALSA requires $\mathcal{O}(n^2)$ space to store the kernel matrix and $\mathcal{O}(n^3)$ effort to invert it. Some recent advances in scalable kernel methods such as random features, divide and conquer techniques, stochastic gradients etc. (Dai et al., 2014; Le et al., 2013; Rahimi & Recht, 2007; 2009; Zhang et al., 2013) can be explored to scale SALSA with n. However, this is beyond the scope of this paper and is left to future work. For this reason, we also limit our experiments to moderate dataset sizes. The goal of this paper is primarily to introduce additive models of higher order, address the combinatorial cost in such models and theoretically demonstrate the improvements in the excess risk.

4. Experiments

We compare SALSA to the following. **Nonparametric models:** Kernel Ridge Regression (KRR), k-Nearest Neighbors (kNN), Nadaraya Watson (NW), Locally Linear/ Quadratic interpolation (LL, LQ), ϵ-Support Vector Regression ($\epsilon-$SVR), ν-Support Vector Regression ($\nu-$SVR), Gaussian Process Regression (GP), Regression Trees (RT), Gradient Boosted Regression Trees (GBRT) (Friedman, 2000), RBF Interpolation (RBFI), M5' Model Trees (M5') (Wang & Witten, 1997) and Shepard Interpolation (SI). **Nonparametric additive models:** Back-fitting with cubic splines (BF) (Hastie & Tibshirani, 1990), Multivariate Adaptive Regression Splines (MARS) (Friedman, 1991), Component Selection and Smoothing (COSSO) (Lin & Zhang, 2006), Sparse Additive Models (SpAM) (Ravikumar et al., 2009) and Additive Gaussian Processes (Add-GP) (Duvenaud et al., 2011). **Parametric models:** Ridge Regression (RR), Least Absolute Shrinkage and Selection (LASSO) (Tibshirani, 1994) and Least Angle Regression (LAR) (Efron et al., 2004). We used software from (Chang & Lin, 2011; Hara & Chellappa, 2013; Jakabsons, 2015; Lin & Zhang, 2006; Rasmussen & Williams, 2006) or from Matlab. In some cases we used our own implementation.

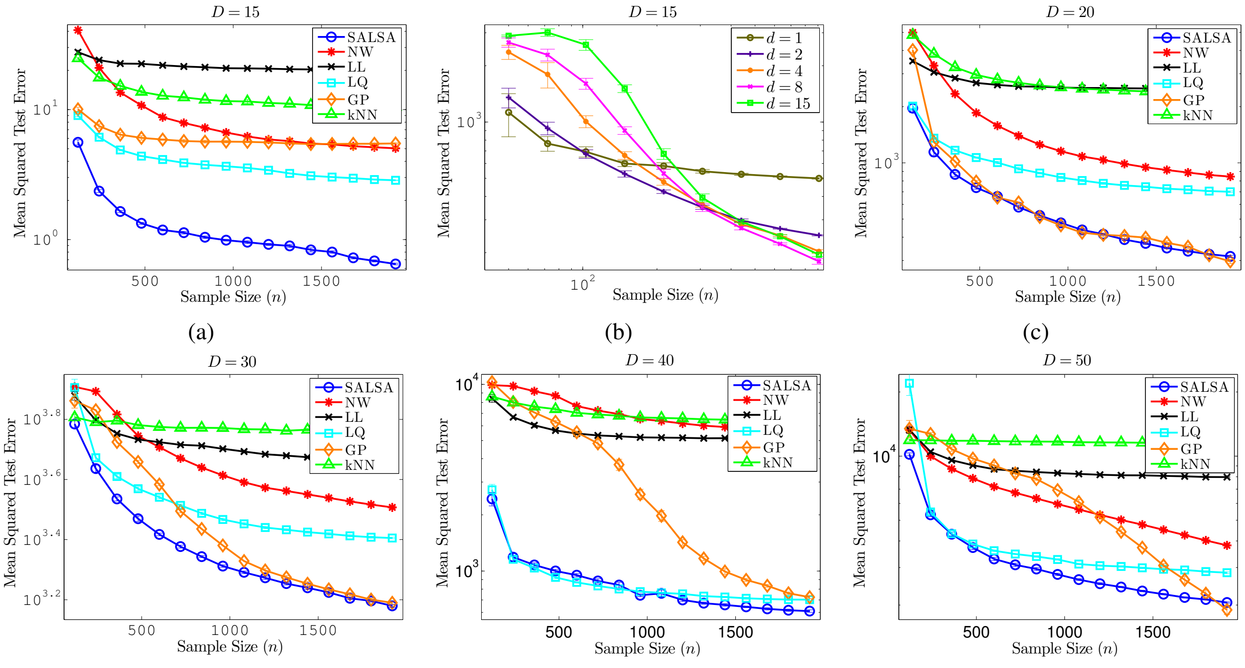
4.1. Synthetic Experiments

We begin with a series of synthetic examples. We compare SALSA to some non-additive methods to convey intuition about our additive model. First we create a synthetic low order function of order $d = 3$ in $D = 15$ dimensions. We do so by creating a d dimensional function f_d and add that function over all $\binom{D}{d}$ combinations of coordinates. We compare SALSA using order 3 and compare against others. The results are given in Figure 1(a). This setting is tailored to the assumptions of our method and, not surprisingly, it outperforms all alternatives.

Next we demonstrate the bias variance trade-offs in using additive approximations on non-additive functions. We created a 15 dimensional (non-additive) function and fitted a SALSA model with $d = 1, 2, 4, 8, 15$ for difference choices of n. The results are given in Figure 1(b). The interesting observation here is that for small samples sizes small d performs best. However, as we increase the sample size we can also increase the capacity of the model by accommodating higher orders of interaction. In this regime, large d produces the best results. This illustrates our previous point that the order of the additive model gives us another way to control the bias and variance in a regression task. We posit that when n is extremely large, $d = 15$ will eventually beat all other models. Finally, we construct synthetic functions in $D = 20$ to 50 dimensions and compare against other methods in Figures 1(c) to 1(f). Here, we chose d via cross validation. Our method outperforms or is competitive with other methods.

4.2. Real Datasets

Finally we compare SALSA against the other methods listed above on 16 datasets. The datasets were taken from the UCI repository, Bristol Multilevel Modeling and the following sources: (Guillame-Bert et al., 2014; Just et al., 2010; Paschou, 2007; Tegmark et al, 2006; Tu, 2012; Wehbe et al., 2014). Table 1 gives the average squared error



(a) (b) (c)

(d) (e) (f)

Figure 1. (a) Comparison of SALSA which knows the additive order of f_* against other methods. (b) Comparison of different choices of d in SALSA. The best d varies with n. (c)-(f) Comparison of SALSA (d chosen via cross validation) with alternatives on synthetic datasets. In all cases, we plot the mean squared prediction error on a test set of 2000 points. All curves are produced by averaging over 10 trials. The error bars are not visible in some curves as they are very small.

on a test set. For the Speech dataset we have indicated the training time (including cross validation for selecting hyper-parameters) for each method. For SALSA we have also indicated the order d chosen by cross validation. See the caption under the table for more details.

SALSA performs best (or is very close to the best) in 5 of the datasets. Moreover it falls within the top 5 in all but two datasets, coming sixth in both instances. Observe that in many cases d chosen by SALSA is much smaller than D, but *importantly* also larger than 1. This observation (along with Fig 1(b)) corroborates a key theme of this paper: while it is true that additive models improve the variance in high dimensional regression, it is often insufficient to confine ourselves to just first order models.

In Appendix D we have given the specifics on the datasets such as preprocessing, the predictors, features etc. We have also discussed some details on the alternatives used.

5. Conclusion

SALSA finds additive approximations to the regression function in high dimensions. It has less bias than first order models and less variance than non-additive methods. Algorithmically, it requires plugging in an additive kernel to KRR. In computing the kernel, we use the Girard-Newton formulae to efficiently sum over a combinatorial number of terms. Our theorems show that the excess risk depends only polynomially on D when f_* is additive, significantly better than the usual exponential dependence of nonparametric methods, albeit under stronger assumptions. Our analysis of the agnostic setting provides intuitions on the tradeoffs invovled with changing d. We demonstrate the efficacy of SALSA via a comprehensive empirical evaluation. Going forward, we wish to use techniques from scalable kernel methods to handle large datasets.

Theorems 3,4 show polynomial dependence on D when f_* is additive. However, these theorems are unsatisfying since in practice regression functions need not be additive. We believe our method did well even on non-additive settings since we could control model capacity via d. In this light, we pose the following open problem: identify suitable assumptions to beat existing lower bounds and prove faster convergence of additive models whose additive order d increases with sample size n. Our Theorem 5 might be useful in this endeavour.

Acknowledgements

We thank Calvin McCarter, Ryan Tibshirani and Larry Wasserman for the insightful discussions and feedback on the paper. We also thank Madalina Fiterau for providing us with datasets. This work was partly funded by DOE grant DESC0011114.

Dataset (D, n)	SALSA (d)	KRR	kNN	NW	LL	LQ	$\epsilon-$SVR	$\nu-$SVR	GP	RT	GBRT
Housing (12, 256)	**0.26241** (9)	0.37690	0.43620	0.38431	0.31219	0.35061	1.15272	0.38600	0.67563	1.06015	0.42951
Galaxy (20,2000)	**0.00014** (4)	0.01317	0.25854	0.30615	0.01676	0.00175	0.65280	0.15798	0.00221	0.02293	0.01405
fMRI (100,700)	**0.80730** (2)	0.86495	0.85645	0.84989	0.91098	1.14079	0.81080	0.81376	0.84766	1.52834	0.87326
Insulin (50,256)	**1.02062** (3)	1.09023	1.15578	1.18070	1.06457	1.35747	1.10725	1.09140	1.22404	1.58009	1.06624
Skillcraft (18,1700)	**0.54695** (1)	0.54803	0.67155	0.73258	0.60581	1.29690	0.71261	0.66311	0.54816	1.08047	0.57273
School (36,90)	1.32008 (2)	1.64315	1.37910	**1.14866**	2.13390	4.79447	1.38173	1.48746	1.64215	1.99244	2.09863
CCPP* (59,2000)	0.06782 (2)	0.08038	0.32017	0.33863	0.07568	0.06779	0.33707	0.094493	0.11128	1.04527	0.06181
Bleeding (100,200)	0.00123 (5)	0.10633	0.16727	0.19284	**2.48e − 6**	0.00614	0.36505	0.03168	6.68e − 6	0.18292	0.19076
Speech (21, 520)	0.02246 (2)	0.03036	0.09348	0.11207	0.03373	0.02404	0.22431	0.06994	0.02531	0.05430	0.03515
Training time	$4.71s$	$0.8s$	$0.18s$	$1.81s$	$4.53s$	$6.80s$	$0.24s$	$27.43s$	$6.34s$	$0.21s$	$5.30s$
Music (90,1000)	0.62512 (3)	0.61244	0.71141	0.75225	0.67271	1.31957	0.75420	**0.59399**	0.62429	1.45983	0.66652
Telemonit (19, 1000)	0.03473 (9)	0.05640	0.09262	0.21198	0.08253	0.18399	0.33902	0.05246	0.03948	**0.01375**	0.04371
Propulsion (15,200)	0.00881 (8)	0.05010	0.14614	0.11237	1.12712	1.12801	0.74511	0.00910	0.00355	0.02341	**0.00061**
Airfoil* (40,750)	0.51756 (5)	0.53111	0.85879	0.86752	0.51877	0.51105	0.64853	0.55118	0.54494	0.45249	**0.34461**
Forestfires(10, 211)	0.35301 (3)	0.28771	0.36571	0.37199	0.35462	12.5727	0.70154	0.43142	0.29038	0.41531	0.26162
Brain (29,300)	0.00036 (2)	0.01239	0.24429	0.22929	1.23412	2.45781	0.27204	0.05556	5e − 14	0.00796	0.00693

	RBFI	M5'	SI	BF	MARS	COSSO	SpAM	Add-GP	RR	LASSO	LAR
Housing (12, 256)	0.64871	0.38256	0.50445	0.64218	0.42379	1.30965	0.81653	0.45656	95.60708	0.44515	0.84410
Galaxy (20,2000)	0.01532	0.00116	0.92189	0.94165	0.00163	0.00153	0.95415	–	0.13902	0.02392	1.02315
fMRI (100,700)	1.38585	1.42795	0.90595	0.86197	0.90850	0.82448	0.88014	–	0.81005	0.81390	0.88351
Insulin (50,256)	1.22404	1.78252	1.20771	1.16524	1.10359	1.13791	1.20345	–	**1.02051**	1.11034	1.22404
Skillcraft (18,1700)	0.81966	1.07195	0.87677	0.83733	**0.54595**	0.55514	0.905445	–	0.70910	0.66496	1.00048
School (36,90)	1.61927	1.72657	1.52374	1.48866	1.44453	1.48046	1.59328	–	1.53416	1.34467	1.64330
CCPP* (59,2000)	1.04257	0.10513	0.97223	0.88084	0.08189	0.96844	**0.06469**	–	0.07641	0.07395	1.04527
Bleeding (100,200)	0.13872	0.00210	0.24918	0.37840	0.00497	0.31362	0.41735	–	0.00001	0.00191	0.43488
Speech (21, 520)	0.03339	0.02843	0.39883	0.36793	**0.01647**	0.34863	0.66009	0.02310	0.07392	0.07303	0.73916
Training time	$0.12s$	$5.70s$	$0.66s$	$27.93s$	$8.11s$	$9.40s$	$76.39s$	$79\,mins$	$0.03s$	$15.94s$	$0.06s$
Music (90,1000)	0.78482	1.28709	0.77347	0.75646	0.88779	0.79816	0.76830	–	0.67777	0.63486	0.78533
Telemonit (19, 1000)	0.02872	0.01491	0.65386	0.84412	0.02400	5.71918	0.86425	–	0.08053	0.08629	0.94943
Propulsion (15,200)	0.05832	0.02341	0.27768	0.56418	0.0129	0.00094	1.11210	0.01435	0.01490	0.02481	10.2341
Airfoil* (40,750)	1.00909	0.46714	0.99413	0.96668	0.54552	0.51782	0.96231	–	0.53187	0.51986	1.00910
Forestfires(10, 211)	0.45773	0.47749	0.78057	0.88979	0.33891	0.37534	0.96944	**0.17024**	0.47892	0.51934	1.05415
Brain (29,300)	0.97815	**2e − 37**	0.81711	0.63700	5e − 31	2e − 6	0.89533	–	4e − 13	0.00089	1.04216

Table 1. The average squared error on the test set for all methods on 16 datasets. The dimensionality and sample size n are indicated next to the dataset. The best method(s) for each dataset are in bold. The second to fifth methods are underlined. For SALSA we have also indicated the order d chosen by our cross validation procedure in parantheses. The datasets with a * are actually lower dimensional datasets from the UCI repository. But we artificially increase the dimensionality by inserting random values for the remaining coordinates. Even though, this doesn't change the function value it makes the regression problem harder.

References

Aronszajn, N. Theory of Reproducing Kernels. *Trans. Amer. Math. Soc.*, 1950.

Bach, Francis R. Consistency of the Group Lasso and Multiple Kernel Learning. *JMLR*, 2008.

Berlinet, Alain and Thomas-Agnan, Christine. *Reproducing kernel Hilbert spaces in Probability and Statistics*. Kluwer Academic, 2004.

Chang, Chih-Chung and Lin, Chih-Jen. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3), 2011.

Dai, Bo, Xie, Bo, He, Niao, Liang, Yingyu, Raj, Anant, Balcan, Maria-Florina F, and Song, Le. Scalable Kernel Methods via Doubly Stochastic Gradients. In *NIPS*, 2014.

Duvenaud, David K., Nickisch, Hannes, and Rasmussen, Carl Edward. Additive Gaussian Processes. In *NIPS*, 2011.

Efron, Bradley, Hastie, Trevor, Johnstone, Iain, and Tibshirani, Robert. Least Angle Regression. *Annals of Statistics*, 32(2):407–499, 2004.

Friedman, Jerome H. Multivariate Adaptive Regression Splines. *Ann. Statist.*, 19(1):1–67, 1991.

Friedman, Jerome H. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, 2000.

Gönen, Mehmet and Alpaydin, Ethem. Multiple Kernel Learning Algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011.

Guillame-Bert, M, Dubrawski, A, Chen, L, Holder, A, Pinsky, MR, and Clermont, G. Utility of Empirical Models of Hemorrhage in Detecting and Quantifying Bleeding. In *Intensive Care Medicine*, 2014.

Györfi, László, Kohler, Micael, Krzyzak, Adam, and Walk, Harro. *A Distribution Free Theory of Nonparametric Regression*. Springer Series in Statistics, 2002.

Hara, Kentaro and Chellappa, Rama. Computationally efficient regression on a dependency graph for human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, 2013.

Hastie, T. J. and Tibshirani, R. J. *Generalized Additive Models*. London: Chapman & Hall, 1990.

Hastie, T. J., Tibshirani, R. J., and Friedman, J. *The Elements of Statistical Learning*. Springer., 2001.

Jakabsons, Gints. Open source regression software for Matlab/Octave, 2015.

Just, Marcel Adam, Cherkassky, Vladimir L., Aryal, S, Mitchell, Tom M., Just, Marcel Adam, Cherkassky, Vladimir L., Aryal, Esh, and Mitchell, Tom M. A neurosemantic theory of concrete noun representation based on the underlying brain codes. *PLoS ONE*, 2010.

Lafferty, John D. and Wasserman, Larry A. Rodeo: Sparse Nonparametric Regression in High Dimensions. In *NIPS*, 2005.

Le, Quoc, Sarlos, Tamas, and Smola, Alex. Fastfood - Approximating Kernel Expansions in Loglinear Time. In *30th International Conference on Machine Learning (ICML)*, 2013.

Lin, Yi and Zhang, Hao Helen. Component selection and smoothing in smoothing spline analysis of variance models. *Annals of Statistics*, 2006.

Lou, Yin, Caruana, Rich, Gehrke, Johannes, and Hooker, Giles. Accurate Intelligible Models with Pairwise Interactions. In *KDD*, 2013.

Macdonald, Ian Grant. *Symmetric functions and Hall polynomials*. Clarendon Press, 1995.

Paschou, P. PCA-correlated SNPs for Structure Identification. *PLoS Genetics*, 2007.

Plate, Tony A. Accuracy versus Interpretability in flexible modeling:implementing a tradeoff using Gaussian process models. *Behaviourmetrika, Interpreting Neural Network Models"*, 1999.

Rahimi, Ali and Recht, Benjamin. Random Features for Large-Scale Kernel Machines. In *NIPS*, 2007.

Rahimi, Ali and Recht, Benjamin. Weighted Sums of Random Kitchen Sinks: Replacing minimization with randomization in learning. In *NIPS*, 2009.

Rasmussen, C.E. and Williams, C.K.I. *Gaussian Processes for Machine Learning*. Cambridge University Press, 2006.

Ravikumar, Pradeep, Lafferty, John, Liu, Han, and Wasserman, Larry. Sparse Additive Models. *Journal of the Royal Statistical Society: Series B*, 71:1009–1030, 2009.

Schölkopf, Bernhard and Smola, Alexander J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

Shawe-Taylor, John and Cristianini, Nello. *Kernel Methods for Pattern Analysis*. Cambridge university press, 2004.

Steinwart, Ingo and Christmann, Andreas. *Support Vector Machines*. Springer, 2008.

Steinwart, Ingo, Hush, Don R., and Scovel, Clint. Optimal Rates for Regularized Least Squares Regression. In *COLT*, 2009.

Tegmark et al, M. Cosmological Constraints from the SDSS Luminous Red Galaxies. *Physical Review*, 74(12), 2006.

Tibshirani, Robert. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society, Series B*, 1994.

Tsybakov, Alexandre B. *Introduction to Nonparametric Estimation*. Springer, 2008.

Tu, Zhidong. Integrative Analysis of a cross-locci regulation Network identifies App as a Gene regulating Insulin Secretion from Pancreatic Islets. *PLoS Genetics*, 2012.

Wang, Yong and Witten, Ian H. Inducing Model Trees for Continuous Classes. In *ECML*, 1997.

Wehbe, L., Murphy, B., Talukdar, P., Fyshe, A., Ramdas, A., and Mitchell, T. Simultaneously uncovering the patterns of brain regions involved in different story reading. *PLoS ONE*, 2014.

Williamson, Robert C., Smola, Alex J., and Schölkopf, Bernhard. Generalization Performance of Regularization Networks and Support Vector Machines via Entropy Numbers of Compact Operators. *IEEE Transactions on Information Theory*, 47(6):2516–2532, 2001.

Xu, Zenglin, Jin, Rong, Yang, Haiqin, King, Irwin, and Lyu, Michael R. Simple and Efficient Multiple Kernel Learning by Group Lasso. In *ICML*, 2010.

Zhang, Tong. Learning Bounds for Kernel Regression Using Effective Data Dimensionality. *Neural Computation*, 17(9):2077–2098, 2005.

Zhang, Yuchen, Duchi, John C., and Wainwright, Martin J. Divide and Conquer Kernel Ridge Regression. In *COLT*, 2013.