Literature Review: Nonparametric Additive Models

Jeffrey Mei

Abstract

Nonparametric additive models are flexible extensions of linear models. They can fit to complex smooth surfaces, but they suffer in high dimensions since they potentially fit to insignificant components. Therefore, a mechanism for filtering out insignificant components is of paramount importance. Since the advent of the LASSO [Tibshirani, 1996], many procedures have applied regularization techniques to instill variable selection capabilities. In this literature review, we cover two approaches that apply soft-thresholding to filter out insignificant components in the context of additive models. Lin and Zhang propose the COSSO (COmponent Selection and Smoothing Operator), which generalizes the LASSO to the context of additive models in order to facilitate component selection [Lin and Zhang, 2006]. Building on this work, [Ravikumar et al., 2008] proposes a variation that generalizes the COSSO and applies a backfitting procedure to facilitate variable selection.

1 Additive Models

Additive models were introduced as a generalization of linear models [Hastie and Tibshirani, 1986]. They take the form

$$Y_i = \sum_{j=1}^{p} f_j(X_{ij}) + \varepsilon_i \tag{1}$$

where ε is often gaussian noise. It is easy to see that if we let $f(X) = X\beta$, then we have the linear model again. In practice, it is common to let $f \in \mathcal{S}^2[0,1]$, the second order Sobolev space on $[0,1]$, defined as the set $\left\{f : f, f' \text{ abs. continuous}, \int_0^1 (f''(x))^2 dx < \infty\right\}$. This is an enormous class of functions that includes polynomials, sin, cos, log, and many more functions, making additive models an incredibly flexible approach to data modeling.

It is worth noting that additive models can be extended to the class of Smoothing-Spline Analysis of Variance (SS-ANOVA) models if interactions are considered as well. That is, SS-ANOVA models take the form

$$Y_i = \sum_{j=1}^{p} f_j(X_{ij}) + \sum_{j<k} f_{jk}(X_{ij}) + ... + \varepsilon_i. \tag{2}$$

While additive models are incredibly flexible, one problem is that they suffer in high dimensions. Often times when p is large, there are components in the model that make little to no contributions to the model performance. However, without filtering out these features, the model will end up fitting to the noise. Therefore,

In this literature review, we will provide an overview of methods used to overcome this issue. In particular we will consider the COmponent Selection and Smoothing Operator (COSSO) [Lin and Zhang, 2006] and the Sparse Additive Model (SpAM) [Ravikumar et al., 2008].

Often times, we can transform a variable to improve model performance by using a spline. However, splines typically introduce more predictors in the model as well. The group lasso will typically attempt to select the entire spline at once. However, the COSSO and SpAM models circumvent the issue entirely by searching through the Hilbert space that contains splines

To build intuition, Figure 1 illustrates a simple case where the complex surface f can be decomposed into two simpler constituent components: a sinusoidal function and a polynomial function.

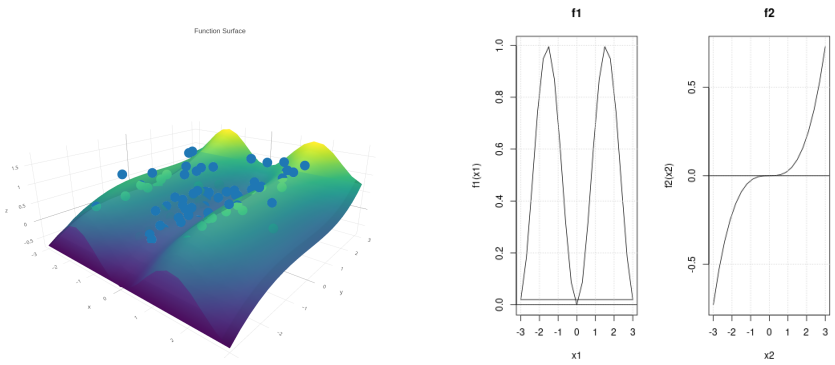

Figure 1: Function Surface

1.1 Reproducing Kernel Hilbert Spaces

It is essential to understand the basics of Reproducing Kernel Hilbert Spaces (RKHS), as they play a significant role in both the COSSO and SpAM procedures.

RKHSs are useful tools in the smoothing spline literature. They offer a feasible approach to access a potentially infinite dimensional class of functions (e.g. 2nd order Sobolev space). For a more detailed approach, refer to [Gu, 2002] and [Nosedal-Sanchez et al., 2012]. Here, we will provide an informal overview of RKHS theory.

A **functional** L is defined as a mapping from a linear space $\mathcal{V}$ to the real numbers $L : \mathcal{V} \to \mathbb{R}$. A functional of particular interest is the **evaluation functional** $[\cdot]$, such that $[x]f = f(x)$. That is, the evaluation functional $[x]$ is equal to evaluating the function f at x.

Theorem 1.1. *Riesz Representation Theorem*

Let $\mathcal{H}$ be a Hilbert space with continuous functional L defined on it. For any $f \in \mathcal{H}$, there exists a unique $g \in \mathcal{H}$ such that $Lf = \langle f, g \rangle$.

If we take $[x]$ to be the evaluation functional, it follows that for any continuous function f, there exists some representer R_x such that $\langle R_x, f \rangle = f(x)$. We will call R_x the **reproducing kernel** of the RKHS $\mathcal{H}_R$.

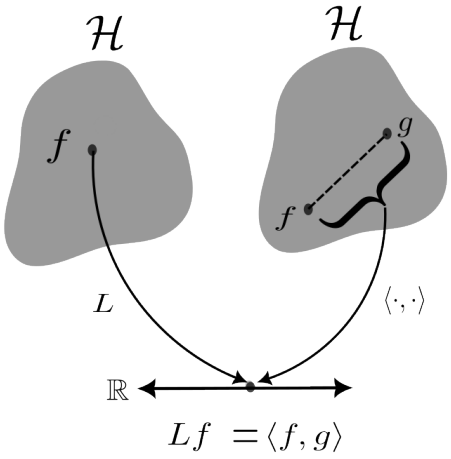

Figure 2: Riesz Representation Theorem

The significance of this result is that it provides two different ways to evaluate a functional. We can evaluate it on the set X with the kernel, or we can project the data onto a different space $\mathcal{H}_R$ and evaluate the functional using the inner product. Most often however, we will evaluate the functional using the kernel because it is typically more computationally convenient.

To find a RKHS, we note that there is a convenient relationship between non-negative definite functions and reproducing kernels.

Theorem 1.2. *RKHS - Non-Negative Definite Relationship*

For every non-negative definite function $R(x, y)$ on X, there exists a unique RKHS $\mathcal{H}_R$ with the reproducing kernel $R(x, y)$. The converse is also true. For every RKHS $\mathcal{H}_R$, there exists a unique non-negative definite function $R(x, y)$ on X.

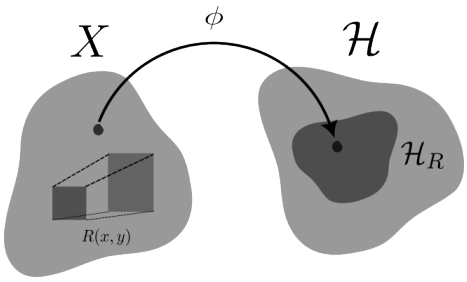

Figure 3: Reproducing Kernel Hilbert Spaces

For the remainder of this paper, we define the inner product to be

$$\langle f, g \rangle = \sum_{v=0}^{m-1} \left(\int_0^1 f^{(v)} dx \right) \left(\int_0^1 g^{(v)} dx \right) + \int_0^1 f^{(m)} g^{(m)} dx. \tag{3}$$

One important consequence of the RKHS framework, is that we are able to represent functions in $\mathcal{H}_R$ by their kernel functions:

$$f(x) = \sum_{\alpha=1}^{p} \theta_\alpha R_\alpha c + b\mathbf{1}_n. \tag{4}$$

By representing functions in $\mathcal{H}_R$ by their kernels, we are able to obtain notions of "similarity" between functions in a richer space $\mathcal{H}_R$ without having to do any computations within that richer space. Instead, our computations remain on the original space X.

2 COSSO: Component Selection and Smoothing

The COSSO is a flexible approach that performs component selection to filter out small to insignificant components. This is similar to the LASSO [Tibshirani, 1996], but instead of penalizing on coefficient size, the COSSO penalizes, in an informal sense, "component size." Roughly speaking, size of a function is described as the squared integral over $[0, 1]$. To develop this fully, we must dive into the theory of Reproducing Kernel Hilbert Spaces (RKHS).

In particular, the problem that we are solving is

$$\hat{f} = \text{argmin}_f \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda_n \sum_{j=1}^{p} \|P^j f\| \tag{5}$$

where the norm in the 2nd order Sobolev space is

$$\|f\| = \left(\int_0^1 f(t)dt \right)^2 + \left(\int_0^1 f'(t)dt \right)^2 + \int_0^1 \left(f''(t) \right)^2 dt. \tag{6}$$

The first term in (5) encourages the optimization problem to fit as closely to the data as possible, but it is penalized by the 2nd term, which penalizes by model complexity.

2.1 Relationship to LASSO

The authors demonstrate that the COSSO and the LASSO are the same. It is important to note that while the COSSO generalizes the LASSO, the interpretation changes. Instead of penalizing on coefficient size, we are penalizing by component size.

2.2 Algorithm

The authors demonstrate that the COSSO can be decomposed into a non-negative garrote and a smoothing spline problem. Since the algorithm is guaranteed to improve its estimate on every iteration, we can just alternate between the non-negative garrote solution and the smoothing spline solution.

The authors note that the original COSSO problem (5) can be reformulated

into the more computationally tractable form

$$\frac{1}{n}\left(y-\sum_{\alpha=1}^{p}\theta_\alpha R_\alpha c-b\mathbf{1}_n\right)^T\left(y-\sum_{\alpha=1}^{p}\theta_\alpha R_\alpha c-b\mathbf{1}_n\right)+\lambda_0\sum_{\alpha=1}^{p}\theta_\alpha c^T R_\alpha c+\lambda\sum_{\alpha=1}^{p}\theta_\alpha. \tag{7}$$

As it turns out, (7) can be broken down further into two constituent sub-algorithms. In particular, when we fix c and b, (7) is reduced into the ridge regression. Similarly, when θ is fixed, (7) reduces into a non-negative garrote.

By fixing c, b, the COSSO reduces to the non-negative garrote, where components are select4ed:

$$\min_{\theta}(z-G\theta)^T(z-G\theta)+n\lambda\sum_{\alpha=1}^{p}\theta_\alpha \tag{8}$$

where $\theta_\alpha \geq 0$ and $z = y-(1/2)n\lambda_0 c-b\mathbf{1}_n$.

Similarly, by fixing θ, we get a problem equivalent to ridge regression, where the functions are smoothed:

$$\min_{c,b}(y-R_\theta c-b\mathbf{1}_n)^T(y-R_\theta c-b\mathbf{1}_n)+n\lambda_0 c^T R_\theta c. \tag{9}$$

Now, with the insight that the COSSO can be broken down into problems with known solutions, the proposed algorithm flips between fixing θ and fixing c and b. In other words, the algorithm flips between the non-negative garrote and ridge regression. We continue until algorithm converges on a solution with a pre-specified error.

Algorithm 1 COSSO

Initialize fix $\theta_\alpha = 1$, $\alpha = 1, ..., p$, $g(\theta, b, c) = 0$

while $g - g' > \varepsilon$ **do**

fix θ apply ridge regression (9)

$c, b \leftarrow \text{argmin}_{c,b}(y-R_\theta c-b\mathbf{1}_n)^T(y-R_\theta c-b\mathbf{1}_n)+n\lambda_0 c^T R_\theta c.$

fix b, c, apply non-negative garrote (8)

$\theta \leftarrow \text{argmin}_\theta(z-G\theta)^T(z-G\theta)+n\lambda\sum_{\alpha=1}^{p}\theta_\alpha$

$g(\theta, b, c) \leftarrow \min_\theta(z-G\theta)^T(z-G\theta)+n\lambda\sum_{\alpha=1}^{p}\theta_\alpha$

end while

However, the authors note that the first iteration makes most of the way to a solution.

3 Sparse Additive Models

Ravikumar et al. (2008) propose the Sparse Additive Model (SpAM), which is similar to the COSSO. Again, the model is proposed is another l_1 penalized approach to add sparsity to the additive model. However, SpAM applies an additional constraint to normalize function size. In doing so, SpAM decouples sparsity and smoothness. Recall that the COSSO penalizes by component sizes, which are functions of both complexity and magnitude. By decoupling sparsity and smoothness, SpAM is more flexible than COSSO.

The optimization problem for SpAM is

$$\min_{g_j \in H_j} \mathbb{E}\left[Y - \sum_{j=1}^{p} \beta_j g_j(X_j)\right]^2 \tag{10}$$

$$\text{subject to: } \sum_{j=1}^{p} |\beta_j| \leq L \tag{11}$$

$$\mathbb{E}\left[g_j^2\right] = 1. \tag{12}$$

where Y is an $n \times 1$ vector representing the outputs to be predicted, X is an $n \times p$ data matrix, $L \geq 0$ is a penalty constraint, and $\mathcal{H}_j$ is a RKHS for $j = 1, ..., p$.

In the LASSO, we penalize the regression coefficients by taking the norm of the β vector. Here, we take the same idea to encourage sparsity, in addition to adding an additional constraint of $\mathbb{E}\left[g_j^2\right] = 1$ to limit the set of functions to search.

We can rewrite the above constraint to be

$$\min_{f_j \in H_j} \mathbb{E}\left[Y - \sum_{j=1}^{p} \beta_j g_j(X_j)\right]^2$$

$$\text{subject to: } \sum_{j=1}^{p} \sqrt{\mathbb{E}\left[f_j^2(X_j)\right]} \leq L.$$

Or equivalently

$$\mathcal{L}(f, \lambda) = \frac{1}{2}\mathbb{E}\left[Y - \sum_{j=1}^{p} f_j(X_j)\right]^2 + \lambda \sum_{j=1}^{p} \sqrt{\mathbb{E}\left[f_j^2(X_j)\right]}. \tag{13}$$

Proof.

$$
\begin{aligned}
f_j(X_j) &= \beta_j g_j(X_j) \\
g_j(X_j) &= f_j(X_j)/\beta_j \\
\mathbb{E}\left[g_j(X_j)^2\right] &= \mathbb{E}\left[f_j(X_j)^2/\beta_j^2\right] = 1 \\
\beta_j^2 &= \mathbb{E}\left[f_j^2(X_j)\right] \\
\beta_j &= \sqrt{\mathbb{E}\left[f_j^2(X_j)\right]} \\
\sum_{j=1}^{p} |\beta_j| &= \sum_{j=1}^{p} \sqrt{\mathbb{E}\left[f_j^2(X_j)\right]} \leq L
\end{aligned}
$$

□

The authors demonstrate that the minimizers can be expressed as the soft-thresholded projection

$$
f_j = \left[1 - \frac{\lambda}{\sqrt{\mathbb{E}\left[P_j^2\right]}}\right]_+ \mathbb{E}\left[R_j | X_j\right]. \tag{14}
$$

Where residuals excluding the contribution of the jth component is $R_j = Y - \sum_{k \neq j} f_k(X_k)$ and the projection from the residuals onto $\mathcal{H}_j$ is

$$
P_j = \mathbb{E}\left[R_j | X_j\right]. \tag{15}
$$

Equation (14) illuminates the inner workings of SpAM. In particular, we can see that the population minimizer is a soft-thresholded projection onto $\mathcal{H}_j$ where the projection P_j attempts to reconstruct the signal using information exclusively in the jth component.

3.1 Algorithm

The problem with our formulation in (14) is that it requires information on the population in $\mathbb{E}\left[P_j^2\right]$ and $\mathbb{E}\left[R_j | X_j\right]$. In most practical situations, we will not know the probability distributions and will thus be unable to obtain the expectations. To bridge this gap, we will produce estimates of the expectations.

We may represent projection of residuals onto $\mathcal{H}_j$ defined in (15) with the transformation of the residuals by the smoothing matrix $\mathcal{S}_j$:

$$
\mathbb{E}\left[P_j\right] \approx \hat{P}_j = \mathcal{S}_j R_j. \tag{16}
$$

Consequently,

$$\sqrt{\mathbb{E}\left[P_j^2\right]} \approx \hat{s}_j = \frac{1}{\sqrt{n}}\|\hat{P}_j\| = \sqrt{\text{mean}(\hat{P}_j^2)}. \tag{17}$$

One natural algorithm to solve the problem (13) is the coordinate descent algorithm. The coordinate descent algorithm is guaranteed to find the global minimum if the function to be optimized can be decomposed into $f(\beta_1, ..., \beta_p) = g(\beta_1, ..., \beta_p) + \sum_{j=1}^{p} h_j(\beta_j)$ where g is both convex and differentiable, and h_j convex but not necessarily differentiable [Hastie et al., 2016]. Obviously, additive models fit neatly within this framework. The authors call this method **backfitting**, which can be thought of as a functional version of coordinate descent.

Algorithm 2 Backfitting

initialize estimates: $f_j = 0$

calculate residuals: $R_j = Y - \sum_{k \neq j} \hat{f}_k(X_k)$

estimate projection $P_j = \mathbb{E}\left[R_j | X\right]$

estimate projection $P_j = \mathcal{S}_j R_j$

apply soft-thresholding $\hat{f}_j = (1 - \lambda/\hat{\mathcal{S}}_j)_+ \hat{P}_j$

calculate norm: $s_j^2 = \frac{1}{n}\sum_{i=1}^{n} \hat{P}_j^2(i)$

center estimate: $\hat{f}_j = \hat{f}_j - \text{mean}(\hat{f}_j)$

4 Numerical Results

We will go over a limited selection of numerical results that the papers go over. Unfortunately, the two papers do not have results that can offer a direct comparison between the two techniques studied in this literature review. Therefore, the selected numerical results are chosen because the author of this report find them interesting.

The authors of the COSSO paper compare their method against an established benchmark study with results shown in Figure 4. They have demonstrated that the COSSO performs incredibly competitively with other popular techniques such as SVM and LDA. Not only does the COSSO perform best in 3 out of the 5 benchmark datasets, but when the COSSO gets beat out, it is only by a narrow margin.

	BUPA	Ionosphere	Pima Indian	Sonar MR	Wisc. BC
n	345	351	768	208	683
d	6	33	8	60	9
COSSO	**71.1** (3.5)	91.1 (3.7)	**77.3** (2.2)	**79.0** (4.5)	97.0 (0.8)
SVM (linear)	67.7 (2.6)	87.1 (3.4)	77.0 (2.4)	74.1 (4.2)	96.3 (1.0)
SVM (RBF)	70.4 (3.2)	95.4 (1.7)	**77.3** (2.2)	75.0 (6.6)	96.4 (1.0)
LS-SVM (linear)	65.6 (3.2)	87.9 (2.0)	76.8 (1.8)	72.6 (3.7)	95.8 (1.0)
LS-SVM (RBF)	70.2 (4.1)	**96.0** (2.1)	76.8 (1.7)	73.1 (4.2)	96.4 (1.0)
LDA	65.4 (3.2)	87.1 (2.3)	76.7 (2.0)	67.9 (4.9)	95.6 (1.1)
QDA	62.2 (3.6)	90.6 (2.2)	74.2 (3.3)	53.6 (7.4)	94.5 (0.6)
Logit	66.3 (3.1)	86.2 (3.5)	77.2 (1.8)	68.4 (5.2)	96.1 (1.0)
C4.5	63.1 (3.8)	90.6 (2.2)	73.5 (3.0)	72.1 (2.5)	94.7 (1.0)
oneR	56.3 (4.4)	83.6 (4.8)	71.3 (2.7)	62.6 (5.5)	91.8 (1.4)
IB	61.3 (6.2)	87.2 (2.8)	73.6 (2.4)	77.7 (4.4)	96.4 (1.2)
Naive Bayes	63.7 (4.5)	92.1 (2.5)	75.5 (1.7)	71.6 (3.5)	**97.1** (0.9)
Majority Rule	56.5 (3.1)	64.4 (2.9)	66.8 (2.1)	54.4 (4.7)	66.2 (2.4)

Figure 4: COSSO tested on benchmark datasets.

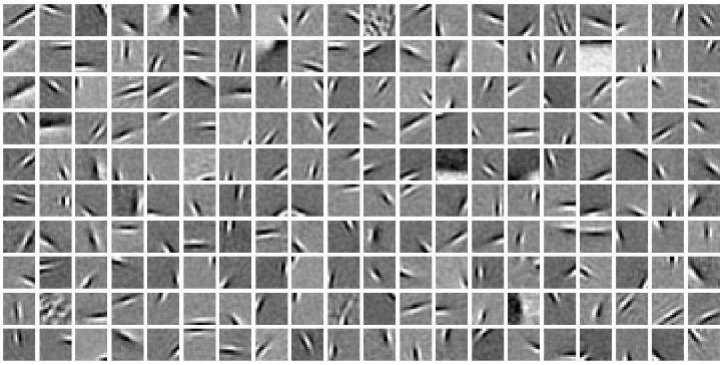

Figure 5: SpAM feature extraction from dataset of images.

While the authors of both techniques extend their methods to the case of binary data, the authors of SpAM also extend their results to enable sparse encoding, a method designed to handle complex data such as images. They note that SpAM extracts edges as demonstrated in Figure 5, suggesting that SpAM does a good job in reducing the dimension of the data and is able to recognize important features.

5 Discussion

It is unfortunate that the authors of SpAM do not make a direct comparison with the COSSO considering their great similarities. While the authors of both papers apply their methods to the Boston data set, the authors of COSSO only report the prediction error, whereas the authors of SpAM only report the selected model. The authors of SpAM appear to have been more interested in how well their method selected variables rather than its overall performance.

Without a direct comparison, it is difficult to determine where one method is superior to the other. The authors of the SpAM claim that the decoupling of sparsity and smoothing in their method provides flexibility than the COSSO. What cost is incurred by adding this flexibility? Perhaps the flexibility comes at no cost. Perhaps the flexibility comes at a great cost. The question goes unanswered for now and may be the subject of a future investigation.

The authors of both papers present interesting ideas that offer solutions to the problem of selecting components from an additive model when p is large. Both techniques are similar as they both rely on l_1 penalization methods and RKHS theory.

References

[Gu, 2002] Gu, C. (2002). *Smoothing Spline ANOVA Models*.

[Hastie and Tibshirani, 1986] Hastie, T. and Tibshirani, R. (1986). Generalized Additive Models. *Statistical Science*, 1(3):297–318.

[Hastie et al., 2016] Hastie, T., Tibshirani, R., Hastie, M. W., Tibshirani, b., and Wainwright, b. (2016). Statistical Learning with Sparsity Monographs on Statistics and Applied Probability 143 143 copy to come from copywriter for review. page 362.

[Lin and Zhang, 2006] Lin, Y. and Zhang, H. H. (2006). COMPONENT SELECTION AND SMOOTHING IN MULTIVARIATE NONPARAMETRIC REGRESSION. *The Annals of Statistics*, 34(5):2272–2297.

[Nosedal-Sanchez et al., 2012] Nosedal-Sanchez, A., Storlie, C. B., Lee, T. C., and Christensen, R. (2012). Reproducing Kernel Hilbert Spaces for Penalized Regression: A Tutorial. *The American Statistician*, 66(1):50–60.

[Ravikumar et al., 2008] Ravikumar, P., Lafferty, J., Liu, H., and Wasserman, L. (2008). SPARSE ADDITIVE MODELS. arXiv: 0711.4555v2.

[Tibshirani, 1996] Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.