Optim Eng (2013) 14:213–224
DOI 10.1007/s11081-013-9215-9

# A distributed algorithm for fitting generalized additive models

**Eric Chu · Arezou Keshavarz · Stephen Boyd**

Received: 11 September 2011 / Accepted: 16 February 2013 / Published online: 29 March 2013


**Abstract** Generalized additive models are an effective regression tool, popular in the statistics literature, that provides an automatic extension of traditional linear models to nonlinear systems. We present a distributed algorithm for fitting generalized additive models, based on the alternating direction method of multipliers (ADMM). In our algorithm the component functions of the model are fit independently, in parallel; a simple iteration yields convergence to the optimal generalized additive model. This is in contrast to the traditional approach of backfitting, where the component functions are fit sequentially. We illustrate the method on different classes of problems such as generalized additive, logistic, and piecewise constant models, with various types of regularization, including those that promote smoothness and sparsity.

**Keywords** Convex optimization · Distributed optimization · Generalized additive models

## 1 Introduction

In this paper we present a distributed algorithm for fitting generalized additive models. Generalized additive models are a powerful regression tool used to model non-

This material is based upon work supported by AFOSR grant FA9550-09-1-0704, by DARPA XDATA FA8750-12-2-0306, and by NASA grant NNX07AEIIA.

E. Chu · A. Keshavarz (✉) · S. Boyd
Information Systems Lab, Electrical Engineering Department, Stanford University, Stanford, CA 94305-9510, USA
e-mail: arezou@stanford.edu

E. Chu
e-mail: echu508@stanford.edu

S. Boyd
e-mail: boyd@stanford.edu

linear regression effects by fitting nonparametric and parametric functions to observed data. Generalized additive models have been well established in the statistics literature (Hastie and Tibshirani 1986; Hastie et al. 2009) and implemented in tools such as GLMNET (Friedman et al. 2010). They have been used effectively in various applications such as biostatistics (Hastie and Tibshirani 1995; Guisan et al. 2002). Their attractiveness arises from their ability to model nonlinearities in data nearly automatically. Furthermore, the traditional linear model (least-squares) is a special case of the generalized additive model, which suggests that where least-squares is commonly used, one might consider using a generalized additive model instead.

A common algorithm used to fit generalized additive models is the backfitting algorithm (Hastie and Tibshirani 1986; Hastie et al. 2009). The drawback, however, is that this algorithm needs to perform computations sequentially. It has been explicitly pointed out that for large data sets, a parallel algorithm for fitting generalized additive models is needed (Hegland et al. 1999). In that paper, the data is split into chunks and the solutions are merged in a single MapReduce operation, yielding a good (but possibly suboptimal) model.

In this paper, we present an algorithm based on the alternating direction method of multipliers (ADMM). ADMM was developed in the 1970s and is closely related to many other algorithms such as dual decomposition, method of multipliers, and Douglas-Rachford splitting (Douglas and Rachford 1956; Everett 1963). For a review of ADMM, see Boyd et al. (2010). ADMM allows generalized additive models to be fit in parallel; instead of distributing the data, ADMM distributes the computation. Thus, our algorithm will coordinate curve-fitting routines to obtain a generalized additive model. Our algorithm is not approximate; it converges to an optimal generalized additive model.

## 2 Generalized additive model

A predictor is a function $\phi$ that attempts to approximate a scalar observation $y$ based on a vector of features $x$; that is, $y \approx \phi(x)$. In this paper, we are interested in the particular class of predictors called generalized additive models, which can be represented as

$$\phi(x) = \psi\left(\sum_i f_i(x_i)\right),$$

where each $f_i \in \mathcal{F}_i$ is a scalar functional from the function class $\mathcal{F}_i$. This formulation generalizes many well-known predictors. For example, in a linear classifier we have $f_i(x_i) = w_i x_i$, and $\psi(u) = \text{sign}(u)$. In a linear regression model, we again take $f_i(x_i) = w_i x_i$, but we take $\psi(u) = u$.

## 3 Fitting generalized additive models

Our goal is to find the best predictor of $m$ observations $y \in \mathbf{R}^m$ based on $N$ feature vectors $x_1, \ldots, x_N \in \mathbf{R}^m$. The predictor has the form $\psi(\sum_{i=1}^N f_i(x_i))$, where

the scalar functions $f_i : \mathbf{R} \to \mathbf{R}$ are applied elementwise to the feature vectors $x_i$. In what follows, we distinguish between the function $f_i$ and the vector $f_i(x_i) \in \mathbf{R}^m$, which consists of the value of $f_i(x_i)$ at the $m$ features. We use the notation $(x_i)_j$ or $(f_i(x_i))_j$ to denote the $j$th element of the vector. For simplicity, $f = (f_1, \ldots, f_N)$ is a vector of functions. The data $y$ and $x_1, \ldots, x_N$ and the function $\psi$ are given; we are to choose the component functions $f_i : \mathbf{R} \to \mathbf{R}$, $i = 1, \ldots, N$.

We can fit a generalized additive model by solving the following optimization problem:

$$\begin{array}{ll} \text{minimize} & L\left(\sum_{i=1}^{N} f_i(x_i)\right) + r(f) \\ \text{subject to} & f_i \in \mathcal{F}_i, \quad i = 1, \ldots, N, \end{array} \tag{1}$$

where the optimization variables are scalar functions $f_i : \mathbf{R} \to \mathbf{R}$, $r(\cdot)$ is a regularization functional, $L$ is a loss function that measures the goodness-of-fit of the predictor $\psi$ at the observed data, and each $\mathcal{F}_i$ is a function space. In general, each $\mathcal{F}_i$ is an infinite-dimensional vector space. However, we are most interested in $f_i$'s that can be represented with a finite vector. This means either that $f_i$ is parametric, or we are only interested in the *value* of $f_i$ at a fixed set of points. (For many practical applications, we can parameterize $f_i$ by discretizing the domain with a finite number of values and work with those instead.) Thus, we are only interested in functions of the form $f_i(\cdot; p_i)$, where $p_i \in \mathbf{R}^n$ is a vector of parameters that specifies $f_i$. For instance, $p_i$ might be a vector of values at certain key points in the domain of $f_i$; the function $f_i$ would be specified by linear or polynomial interpolation through these key points.

Subsequently, we instead solve the *finite*-dimensional problem

$$\text{minimize } L\left(\sum_{i=1}^{N} f_i(x_i; p_i)\right) + r(p_1, \ldots, p_N), \tag{2}$$

where $p_i \in \mathbf{R}^{n_i}$ are the parameters that specify each function $f_i$, and the optimization variables are the values of $f_i$ at each $x_i$ and the parameters $p_i$.

We will consider the case where $L$ is a sum of losses corresponding to the mismatch between the model and each sample $j$, i.e., $L(v) = \sum_{j=1}^{m} l_j(v_j)$, and $r(p_1, \ldots, p_N)$ is decomposable across the features, i.e.,

$$r(p_1, \ldots, p_N) = \sum_{i=1}^{N} r_i(p_i).$$

Our goal is to find function values $f^\star = (f_1(x_1; p_1^\star), \ldots, f_N(x_N; p_N^\star))$ and function parameters $p^\star = (p_1^\star, \ldots, p_N^\star)$ that best explain the observed data as a solution to (2). If $l_j$ is convex for $j = 1, \ldots, m$ and $r_i$ is convex for $i = 1, \ldots, N$, then (2) is a convex optimization problem and can be solved efficiently (Boyd and Vandenberghe 2004).

## 4 Distributed generalized additive models

We propose a distributed method for fitting generalized additive models using the alternating direction method of multipliers (ADMM); specifically, we use the sharing formulation (Boyd et al. 2010).

We introduce dummy variables $z_1, \ldots, z_N \in \mathbf{R}^m$ (representing the values of the features) and write the problem of fitting a generalized additive model as

$$\begin{aligned} &\text{minimize } L\left(\sum_{i=1}^{N} z_i\right) + \sum_{i=1}^{N} r_i(p_i) \\ &\text{subject to } z_i = f_i(x_i; p_i), \quad i = 1, \ldots, N. \end{aligned} \tag{3}$$

Applying the method of Boyd et al. (2010), the ADMM algorithm for fitting generalized additive models becomes

$$p_i^{k+1} := \operatorname{argmin} r_i(p_i) + \rho/2 \left\| f_i(x_i; p_i) - f_i\left(x_i; p_i^k\right) + \overline{f}^k - \overline{z}^k + u^k \right\|_2^2, \quad i = 1, \ldots, N \tag{4}$$

$$\overline{z}^{k+1} := \operatorname*{argmin}_{\overline{z}} L(N\overline{z}) + (N\rho/2) \left\| \overline{z} - u^k - \overline{f}^{k+1} \right\|_2^2 \tag{5}$$

$$u^{k+1} := u^k + \overline{f}^{k+1} - \overline{z}^{k+1}, \tag{6}$$

with $\overline{z}^{k+1} = (1/N)\sum_{i=1}^N z_i^{k+1}$ and $\overline{f}^{k+1} = (1/N)\sum_{i=1}^N f_i(x_i; p^{k+1})$. The standard convergence theory for ADMM tells us that $p_i^k$ will converge to optimal and $z_i^k$ will converge to the corresponding function values at the data $x_i$.

Note that $z_i^{k+1}$ is not explicitly computed; we instead work with $\overline{z}^{k+1}$, the average value of $z_i^{k+1}$. The $p_i$-update fits a function that minimizes the total square error of the function evaluated at the data points in the vector $x_i$. The kind of function fit depends on the choice of $r_i$. Since each $p_i$-update is independent from the others, they can be carried out in $N$ parallel computations.

A common choice for $r_i$ is the $\ell_2$ penalty for continuous functions,

$$r_i(p_i) = \lambda_i \int f_i''(t; p_i)^2 dt.$$

In this case, it is well known that the optimal solution to (4) is a cubic spline, with knots at the data points (Reinsch 1967). Thus, $p_i$ is a vector of coefficients for each data point $(x_i)_j$. The $p_i$-update step can be carried out efficiently by fitting a cubic spline to the data points and can be done in parallel.

For the $\overline{z}$-update, we only need to evaluate the fitted $f_i$'s (specified by their parameters, $p_i$) at $x_i$. We will average the vectors $f_i(x_i; p_i^{k+1})$ to form $\overline{f}^{k+1} \in \mathbf{R}^m$, which is then used in the $\overline{z}$-update step. The $\overline{z}$-update step involves solving a finite dimensional optimization problem (since $\overline{z} \in \mathbf{R}^m$). Once the algorithm terminates, each $p_i$-update block contains the parameters for $f_i$, which can be used on new data to perform predictions.

This approach allows us to perform $N$ function fitting routines *in parallel* and coordinate them via loss functions to produce a solution to (2). This is in contrast to an algorithm such as backfitting, which is essentially sequential: the $p_i$-update depends on $p_1^{k+1}, \ldots, p_{i-1}^{k+1}$ and $p_{i+1}^k, \ldots, p_N^k$, the previously fitted parameters.

### 4.1 Computing the $\overline{z}$-update

It is important to note that for any convex loss function $L$, finding a generalized additive model also requires computing the proximal operator for $L$, defined as

$$\mathbf{prox}_L(v) = \underset{x}{\operatorname{argmin}}\big(L(x) + (\mu/2)\|x - v\|_2^2\big).$$

The $\overline{z}$-update (5) can be expressed as

$$\overline{z}^{k+1} := (1/N)\mathbf{prox}_L\big(Nu^k + N\overline{f}^{k+1}\big),$$

with a choice of $\mu = \rho/N$. Often, there are closed-form solutions for the proximal operator for common choices of $L$, such as quadratic, logistic, or hinge losses. Even when closed-form solutions do not exist or the loss function is nonsmooth, the proximal operator is strongly convex and Newton's method (or fast, first-order methods) can be employed to find its value (Becker et al. 2011b).

### 4.2 Stopping criterion

The stopping criterion for ADMM is as follows: we stop when both the primal and dual residual of problem (3) are small. Following the derivation in (Boyd et al. 2010), the primal residual norm $\|r^k\|_2$ and dual residual norm $\|s^k\|_2$ are

$$\big\|r^k\big\|_2 = \left(\sum_{i=1}^N \big\|f_i\big(x_i; p_i^k\big) - z_i^k\big\|_2^2\right)^{1/2}, \qquad \big\|s^k\big\|_2 = \rho\left(\sum_{i=1}^N \big\|z_i^k - z_i^{k-1}\big\|_2^2\right)^{1/2}.$$

Since $z_i^k = f_i(x_i; p_i^k) + \bar{z}^k - \bar{f}^k$, the residual norms simplify to

$$\big\|r^k\big\|_2 = \sqrt{N}\big\|\bar{f}^k - \bar{z}^k\big\|_2,$$

$$\big\|s^k\big\|_2 = \rho\left(\sum_{i=1}^N \big\|\big(f_i\big(x_i; p_i^k\big) - f_i\big(x_i; p_i^{k-1}\big)\big) + \big(\overline{z}^k - \overline{z}^{k-1}\big) - \big(\overline{f}^k - \overline{f}^{k-1}\big)\big\|_2^2\right)^{1/2}.$$

We terminate ADMM when both the primal and dual residual norms are smaller than some desired tolerance.

## 5 Examples

We will now consider the specific case of additive linear models and additive logistic models. We will also explain how we can apply regressor selection to the additive

models with a slight change in our algorithm. Finally, we will also consider piecewise constant models.

In all cases, where possible, we verified that backfitting and ADMM obtain the same (if not similar) solutions. Furthermore, we observed that ADMM (with a suitable choice of $\rho$) and backfitting require comparable iterations to converge to a similar accuracy; this is not surprising since both methods are first-order methods to solve the optimization problem (2). We define an 'iteration' to mean a single pass over *all* the component functions and computing their fits to the data. The main difference with backfitting is that our ADMM approach can fit the component functions *in parallel* while backfitting requires fitting them *in sequence*.

Since both approaches require fitting functions to the data, each iteration of ADMM—if fully parallelized—is dominated by the *maximum* cost of fitting any function while each iteration of backfitting is dominated by the *total* cost of fitting all the functions. Thus, without sacrificing accuracy, ADMM allows generalized additive models to be fit in parallel and (almost) a factor of $N$ times faster than backfitting.

### 5.1 Additive linear models

Although we do not present any numerical examples for additive linear models, we present the algorithm here for completeness.

Consider an additive linear model of the form

$$y \approx \sum_{i=1}^{N} f_i(x_i; p_i),$$

where the functions $f_i$ are to be estimated, and $y \in \mathbf{R}^m$ is a vector of observations.

We will take the loss function

$$L\left(\sum_{i=1}^{N} f_i(x_i; p_i)\right) = (1/2)\left\| y - \sum_{i=1}^{N} f_i(x_i; p_i)\right\|^2,$$

and we will use the $\ell_2$ penalty for continuous functions for the regularization function. This reduces the $p_i$-update to the fitting of cubic splines.

Because the loss function $L$ is quadratic, the $\overline{z}$-update can be expressed analytically:

$$\overline{z}^{k+1} = \frac{\rho(u^k + \overline{f}^{k+1}) + y}{N + \rho},$$

where $\overline{f}^{k+1} = (1/N)\sum_{i=1}^{N} f_i(x_i; p_i^{k+1})$.

Thus, the process of fitting generalized additive models to data $y$ and $x_1, \ldots, x_N$ reduces to alternating between fitting cubic splines in parallel and averaging the resulting function values.

### 5.2 Additive logistic models

We now consider an additive logistic model, where the observations are binary random variables $y \in \{0, 1\}^m$ with

$$\mathbf{Prob}(y_j = 1) = \frac{\exp(\sum_{i=1}^N (f_i(x_i; p_i))_j)}{1 + \exp(\sum_{i=1}^N (f_i(x_i; p_i))_j)}.$$

A common algorithm to solve this problem is iteratively reweighted backfitting which is presented in (Hastie et al. 2009; Friedman et al. 2010). The function parameters are fit sequentially (with weights) until a desired tolerance is achieved. Here, we show how our algorithm presented in Sect. 4 can be used to solve this problem with a parallel algorithm.

The loss function used to fit this model is the negative log-likelihood given by

$$L\left(\sum_{i=1}^N f_i(x_i; p_i)\right) = \sum_{j=1}^m \log\left(1 + \exp\left(\sum_{i=1}^N \big(f_i(x_i; p_i)\big)_j\right)\right) - \sum_{j=1}^q \sum_{i=1}^N \big(f_i(x_i; p_i)\big)_j,$$

where $q = \sum_{j=1}^m y_j$ is the number of positive samples. We want $f_i$ to be smooth, so we again use the $\ell_2$ penalty for regularization.

The $p_i$-updates are the same as in the additive linear model example—they fit cubic splines. However, since $L$ is the logistic loss function, the $\overline{z}$-update becomes

$$\overline{z}^{k+1} := \underset{\overline{z}}{\operatorname{argmin}} \left(\sum_{j=1}^m \log\big(1+\exp\big(N(\overline{z})_j\big)\big) - \sum_{j=1}^q N(\overline{z})_j\right) + (N\rho/2)\big\|\overline{z} - u^k - \overline{f}^{k+1}\big\|_2^2,$$

where $(\overline{z})_i$ is the $i$th component of the vector $\overline{z}$. Note that the $\overline{z}$-update is strongly convex and completely separable across the samples; so it can be solved efficiently. This extends easily to the multi-class logistic regression (also known as softmax) models; the only difference in the softmax model would be that there would be one set of function parameters for each class.

*Numerical instance* We use the data from the spam example in §9.1 of Hastie et al. (2009). The spam data comes from the UCI machine learning repository. The variable $y$ denotes whether a sample is email (0), or spam (1). There are 57 predictors: 48 of them are based on frequency of word appearances in a message (such as `free`), 6 are based in frequency of character appearances (such as `!`), and the last three predictors are average length, longest length and sum of all lengths of uninterrupted sequence of capital letters.

Figure 1 shows the predictors fitted using our algorithm. These agree with the predictors found via iteratively reweighted backfitting. Again, the advantage of our implementation over iteratively reweighted backfitting is that it can fit the parameters $p_i$ in parallel.

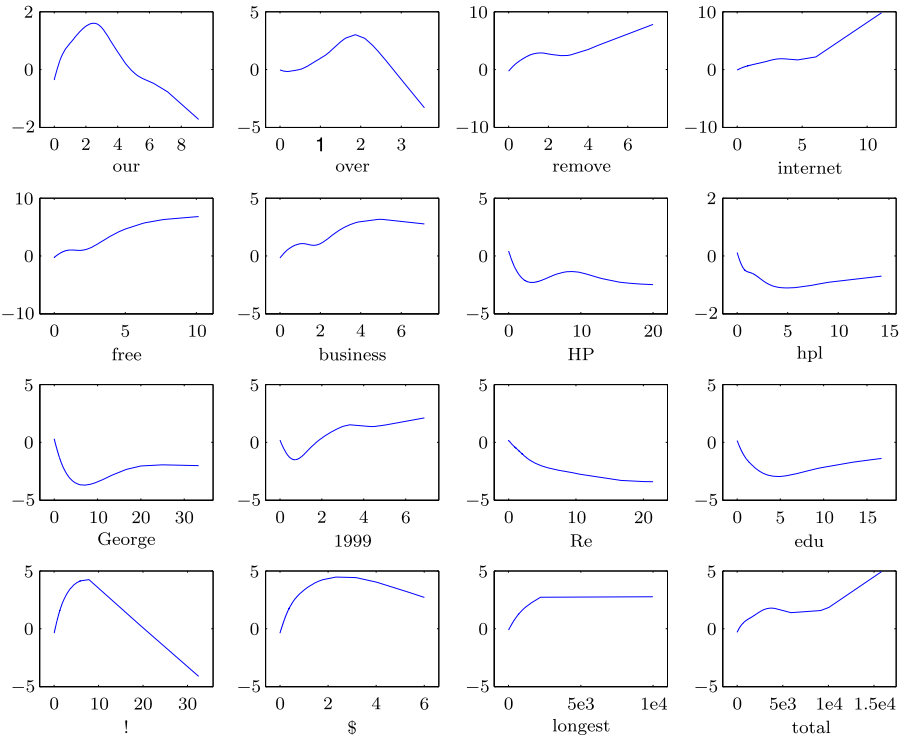


**Fig. 1** Predictors for Hastie et al. (2009) spam example fitted using our algorithm

### 5.3 Additive logistic models with regressor selection

Often, we would like to automatically select a subset of the predictors $x_i$ that best explain our observation $y$. When $f_i$ are linear functions, solving the Lasso problem (Tibshirani 1996) is a good heuristic for choosing the relevant predictors $x_i$ that explain our observation $y$.

Similarly, we would like to choose a sparse subset of $x_i$ in the additive model that best explain $y$. If a predictor $x_i$ is not used, then $f_i = 0$. One approach might be based on iteratively reweighted backfitting and the shrinkage operator, which has been implemented in GLMNET (Friedman et al. 2010). We stress that GLMNET is, in essence, a sequential algorithm; an ADMM-based solution results in a parallel algorithm.

To encourage sparsity among the $f_i$'s, we use the sum-of-norms penalty on $f_i(x_i; p_i)$ to obtain a sparse selection of $x_i$'s (Yuan and Lin 2006; Zhao et al. 2009). This means that we would augment the regularization functional by $\mu_i \| f_i(x_i; p_i) \|$, where the norm could by any norm on $\mathbf{R}^m$. Note that the added norm term is a regularization on the function values at the sample points, and not on the function itself.

For this example, we will take the logistic loss function as in Sect. 5.2, and we will use the $\ell_2$ regularization functional augmented with the $\ell_2$ norm of the function values, i.e., $r_i(p_i) = \int f_i''(t; p_i)^2 dt + \| f_i(x_i; p_i) \|_2$. Note that in this case, if

the $p_i$-update step returns an all zero (parameter) vector, we can conclude that the corresponding cubic spline with minimum total curvature is the zero function.

The $\overline{z}$-updates stay the same as the Sect. 5.2. However, with the added regularization term, the $p_i$-update becomes

$$p_i^{k+1} := \operatorname{argmin} \lambda_i \int f_i''(t; p_i)^2 dt + \mu_i \left\| f_i(x_i; p_i) \right\|_2 + \rho/2 \left\| f_i(x_i; p_i) - f_i\left(x_i; p_i^k\right) + \overline{f}^k - \overline{z}^k + u^k \right\|_2^2.$$

The $p_i$-update involves solving a convex optimization problem, which can be solved in a number of ways. However, we choose to solve the optimization problem involved in the $f_i$-step using ADMM as well. This allows us to decompose the computation by having a dedicated prox operator for $\|f_i(x_i; p_i)\|_2$ (which is employed using group Lasso), and reuse our prox operator for the smoothness penalty from Sect. 5.2.

*Numerical instance* We use the data from the spam example, as in Sect. 5.2. We will first run the algorithm to find a sparse set of features, and we chose 16 features that are most relevant, and then re-fit the logistic model using the chosen features.

Figure 2 shows the predictors refitted using the smaller subset of features. This classifier uses fewer than 30 % of the original predictors while giving a minimal (<5 %) increase in classification error.

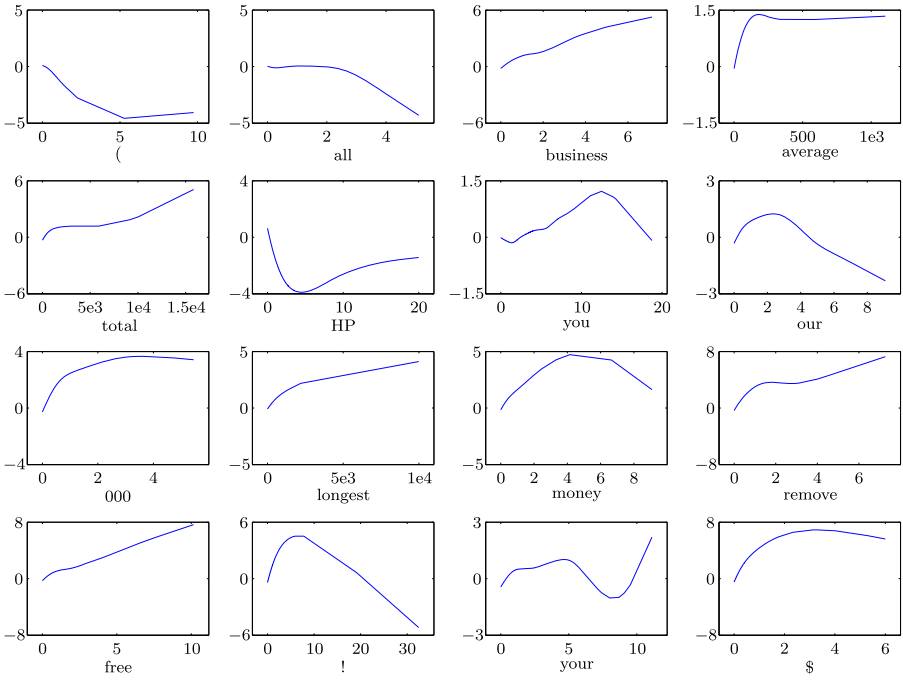


**Fig. 2** The top 16 predictors in the spam example of Hastie et al. (2009) selected by regressor selection, fitted using our algorithm

### 5.4 Additive piecewise-constant models

We again consider a logistic loss function but this time restrict our choice functions to be monotone increasing piecewise-constant functions. This choice of function arises in medical classification problems, where it is desirable to give some interpretation on the resulting predictors (Bottomley et al. 2011). Piecewise-constant predictors automatically bin features in to different levels allowing for simple interpretations of high- or low-risk bins.

The predictors will be modeled as piecewise-constant functions of the regressors, so that

$$f_i(x;\, p_i) = \sum_{j=1}^{m} (p_i)_j \mathcal{I}_{\{x \geq (x_i)_j\}}(x),$$

where the function parameters $p_i$ define the height of the function at each interval and $\mathcal{I}_S(x)$ is the indicator function of the set $S$,

$$\mathcal{I}_S(x) = \begin{cases} 1 & x \in S, \\ 0 & \text{otherwise.} \end{cases}$$

Since we would like to minimize the number of bins used for classification, we instead use a heuristic to promote sparsity. To shape $f_i^\star$, we use a weighted $\ell_1$ regularization on the parameters

$$r_i(p_i) = \lambda_i \| W_i p_i \|_1$$

where $\lambda_i \in \mathbf{R}_+$ and

$$W_i = \operatorname{diag}\left( \frac{1}{(x_i)_2 - (x_i)_1}, \ldots, \frac{1}{(x_i)_m - (x_i)_{m-1}} \right)$$

with the constraints that $p_i \succeq 0$. Note that changing the desired properties of the fitted function only requires that we change how (4) is computed. The $p_i$-update in ADMM can be done by solving the weighted $\ell_1$ problem with nonnegative constraints,

$$\begin{array}{l} \text{minimize } \lambda_i \| W_i p_i \|_1 + \rho/2 \left\| f_i(x_i;\, p_i) - f_i^k\left(x_i;\, p_i^k\right) - \overline{f}^k - \overline{z}^k + u^k \right\|_2^2 \\ \text{subject to } p_i \succeq 0. \end{array}$$

This problem can be solved via standard $\ell_1$ minimization packages such as `l1_ls` or `NESTA` (Kim et al. 2007; Becker et al. 2011a). Without the nonnegativity constraint, the update can be done analytically via a shrinkage operator.

*Numerical instance* We again use data provided by Hastie et al. (2009). The dataset consists of nine predictors used to predict coronary heart disease. These predictors are systolic blood pressure, cumulative tobacco consumption, ldl cholesterol levels, adipose tissue, family history, type-A behavior, body-mass index, alcohol consumption, and age. We fit monotone increasing piecewise constant functions to each predictor,

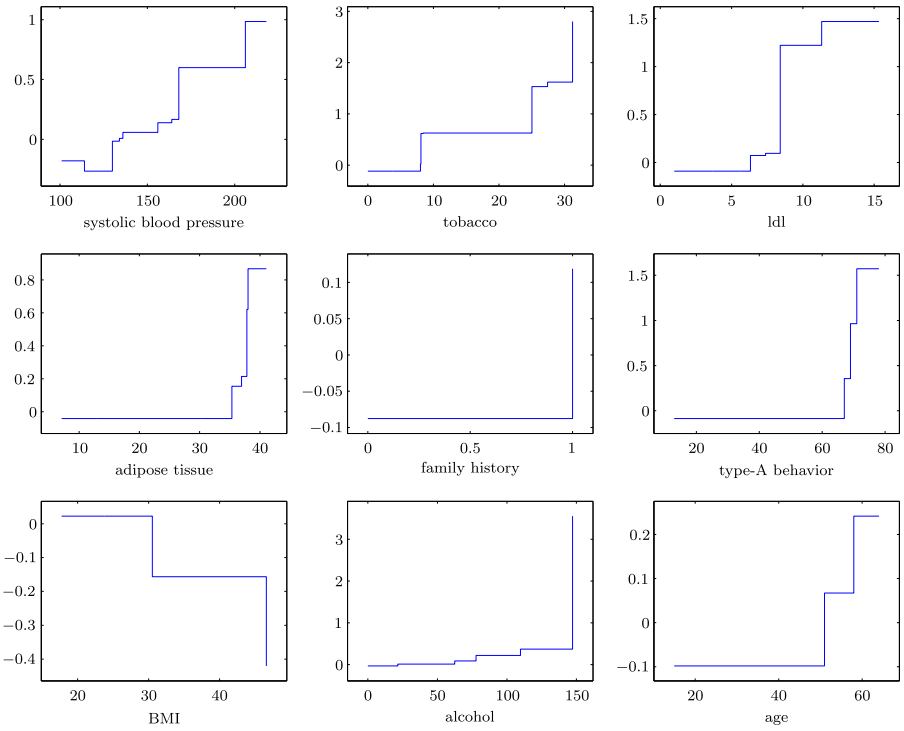


**Fig. 3** Piecewise constant predictors for heart disease classifier

except systolic blood pressure and body-mass index, to obtain interpretable predictors for coronary heart disease.

Figure 3 shows the predictors with $\lambda_1 = 5$ (corresponding to systolic blood pressure), $\lambda_7 = 0.6$ (corresponding to body-mass index) and $\lambda_2 = \cdots = \lambda_6 = \lambda_8 = \lambda_9 = 0.1$ chosen subjectively to provide the most interpretable results. For instance, the age predictor is divided in to three distinct levels which can be interpreted as low-risk up to age 55, medium-risk up to age 60, and high-risk after 60 years of age.

## 6 Conclusion

We can think of generalized additive models as an extension of the simple regression models to nonlinear problems, and as a result, generalized additive models are applicable to many domains, from machine learning to health diagnostic problems. Fitting a generalized additive model can be a computationally intensive task, especially if we have a large number of features. In this paper we have presented a distributed approach for fitting generalized additive models using the alternating direction method of multipliers (ADMM). This approach enables the parallel use of specialized function fitters to fit models of great complexity and in a distributed fashion. We show the application of our method to linear and logistic additive models. We have also

demonstrated how we can promote certain properties in the fitted model, such as sparsity and interpretability.

## References

Becker S, Bobin J, Candès EJ (2011a) NESTA: a fast and accurate first-order method for sparse recovery. SIAM J Imaging Sci 4(1):1–39

Becker S, Candès EJ, Grant M (2011b) Templates for convex cone problems with applications to sparse signal recovery. Math Program Comput 3(3)

Bottomley C, Van Belle V, Pexsters A, Papageorghiou A, Mukri F, Kirk E, Van Huffel S, Timmerman D, Bourne T (2011) A model and scoring system to predict outcome of intrauterine pregnancies of uncertain viability. Ultrasound Obstet Gynecol 37(5):588–595

Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge

Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2010) Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn 3(1):1–122

Douglas J, Rachford HH (1956) On the numerical solution of heat conduction problems in two and three space variables. Trans Am Math Soc 82:421–439

Everett H (1963) Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. Oper Res 11(3):399–417

Friedman JH, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. J Stat Softw 33(1):1–22

Guisan A, Edwards T, Hastie T (2002) Generalized linear and generalized additive models in studies of species distributions: setting the scene. Ecol Model 157:89–100

Hastie T, Tibshirani R (1986) Generalized additive models. Stat Sci 1(3):297–318

Hastie T, Tibshirani R (1995) Generalized additive models for medical research. In: Encyclopedia for biostatistics, vol 4, pp 187–196

Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning: data mining, inference and prediction. Springer, Berlin

Hegland M, McIntosh I, Turlach B (1999) A parallel solver for generalised additive models. Comput Stat Data Anal 31:377–396

Kim S-J, Koh K, Lustig M, Boyd S, Gorinevsky D (2007) An interior-point method for large-scale $\ell_1$-regularized least squares. IEEE J Sel Top Signal Process 1(4):606–617

Reinsch CH (1967) Smoothing by spline functions. Numer Math 10:177–183

Tibshirani R (1996) Regression shrinkage and selection via the lasso. J R Stat Soc, Ser B, Stat Methodol 58(1):267–288

Yuan M, Lin Y (2006) Model selection and estimation in regression with grouped variables. J R Stat Soc, Ser B, Stat Methodol 68:49–67

Zhao P, Rocha G, Yu B (2009) The composite absolute penalties family for grouped and hierarchical variable selection. Ann Stat 37(6A):3468–3497