Trading Interpretability for Accuracy: Oblique Treed Sparse Additive Models

Jialei Wang
University of Chicago
jialei@uchicago.edu

Ryohei Fujimaki
NEC Laboratories America
rfujimaki@nec-labs.com

Yosuke Motohashi
NEC Corporation
y-motohashi@bk.jp.nec.com

Model interpretability has been recognized to play a key role in practical data mining. Interpretable models provide significant insights on data and model behaviors and may convince end-users to employ certain models. In return for these advantages, however, there is generally a sacrifice in accuracy, i.e., flexibility of model representation (e.g., linear, rule-based, etc.) and model complexity needs to be restricted in order for users to be able to under- stand the results. This paper proposes oblique treed sparse additive models (OT-SpAMs). Our main focus is on developing a model which sacrifices a certain degree of interpretability for accuracy but achieves entirely sufficient accuracy with such fully non-linear models as kernel support vector machines (SVMs). OT-SpAMs are instances of region-specific predictive models. They divide feature spaces into regions with sparse oblique tree splitting and assign local sparse additive experts to individual regions. In order to maintain OT-SpAM interpretability, we have to keep the overall model structure simple, and this produces simultaneous model selection issues for sparse oblique region structures and sparse local experts. We address this problem by extending factorized asymptotic Bayesian inference. We demonstrate, on simulation, benchmark, and real world datasets that, in terms of accuracy, OT-SpAMs outperform state-of-the-art interpretable models and perform competitively with kernel SVMs, while still providing results that are highly understandable.

Keywords

Interpretable Model, Model Selection, Sparseness

1. INTRODUCTION

Model interpretability has been recognized to play a key role in practical data mining. Interpretable models provide significant insights on data and model behaviors and may convince end-users to employ certain models. It is well-recognized that, despite the dramatic evolution of machine learning approaches, such as kernel machines [41, 45], boosting [13], random forests [3], and deep neural networks [19, 24], simple models, like linear regressions or decision trees, are still preferred in such applications as market- ing, medical analytics, and science, for which understanding phenomena behind data are more important for end users than sim- ply accurate prediction. In return for advantages in interpretability, however, there is generally a sacrifice in accuracy since flexibility of model representation (e.g., linear, rule-based, etc.) and model complexity need to be restricted in order for users to be able to understand the results.

There are two key concepts in discussions on the issue of model interpretability: *1) model representation* and *2) model complexity*. For the former, linear models (e.g., generalized linear mod- els (GLMs) [31]) and decision trees (e.g., classification and regression tree (CART) [4]) may be considered to be the most interpretable. Although the simplicity of their model representations contributes significantly to end-user understanding, it also limits their predictive ability. For the latter, feature sparseness is a key concept in improving interpretability for linear models; i.e., selecting a small number of key features makes understanding models a lot easier. Also, whiles deep rule chains for decision trees might improve predictive accuracy for complex data, it makes the rule structures hard to understand. The trade-off between accuracy and interpretability remains an important issue.

This paper proposes oblique treed sparse additive models (OT-SpAMs), which provide more flexible representation than linear models and decision trees (and, therefore, sacrifice a certain degree of interpretability.) While offering the same accuracy as that of such fully non-parametric models as kernel support vector machines (KSVMs), they still maintain easily-interpretable model structures. OT-SpAMs are instances of *region-specific predictive models*, which consist of region specifiers and region-specific predictors; the specifiers divide feature spaces into disjoint subspaces (regions), and individual predictors perform predictions in corresponding subspaces (as we note in Section 2, region-specific predictive models unify the above-described two families of interpretable models). OT-SpAMs employ an oblique treed split model as a region specifier and sparse additive models as individual region-specific predictors.

As we have noted above, controlling model complexity is an important issue for maintaining model interpretability. For OT-SpAMs, tree structures (the depth of tree, the number of regions, etc.), feature selection for oblique region splitting, and feature selection for local sparse experts must be determined simultaneously. We address this challenging model selection issue by utilizing factorized asymptotic Bayesian (FAB) inference [11, 14]. Through EM-like iterative optimization, we are able to automatically obtain compact and interpretable OT-SpAMs. We demonstrate, on simulation, benchmark, and real world datasets, that, in terms of accuracy, OT-SpAMs outperform state-of-the-art interpretable models

KDD '15, August 11 - 14, 2015, Sydney, NSW, Australia.

DOI: http://dx.doi.org/10.1145/2783258.2783407.

and perform competitively with kernel SVMs, while still providing results that are highly understandable.

The rest of this paper is organized as follows. Section 2 provides literature reviews of region-specific predictive models. OT-SpAMs and the proposed learning algorithm are presented in Sections 3 and 4, respectively. Simulation studies (Section 5) and benchmark evaluations (Section 6) quantitatively show advantages of OT-SpAMs, and we demonstrate results on real world POS (point-of-sales) data in Section 7.

2. LITERATURE REVIEW

This section focuses mainly on region-specific predictive models. Table 1 summarizes characteristics of region specific models, which are described below. A general and broader survey of interpretable models can be found in [12].

One of the most naive examples is a linear model, which has only one global region and employs a linear prediction model as the region-specific predictor. Some previous studies [18, 21] have argued that oblique hyperplanes of linear models might be hard to understand for end-users. Feature sparseness is a key concept in trying to mitigate this issue, i.e., selecting a small number of key features makes understanding models a lot easier. To obtain sparse linear models (SLMs), various approaches, including convex methods (e.g. Lasso [37], L_1-regularized logistic regression [45]) and greedy optimization (e.g., orthogonal matching pursuit [29, 38]), have been proposed, though their primary focus is on model generalization (to mitigate over-fitting), rather than on enhancing model interpretability. Sparse additive models (SAMs) [20, 32, 34] introduce *feature-wise nonlinearity* to improve accuracy. By restricting nonlinearity in individual features (i.e. ignoring nonlinear interactions among features), we can still visualize their feature-wise (but nonlinear) contributions and get insights from SAMs. Variants of SAMs (kernel density logistic regression (DLR) [6] and fast flux discriminant (FFD) [7]) have been proposed as accurate and interpretable models in recent KDD conferences, and research in this direction has become a topic of intense interest in the community.

Decision trees, such as CART, have tree-structured region specifiers and performs prediction using constant values in individual regions (a.k.a. piecewise constant predictors). Oblique decision trees [33] extend region specifiers from single-feature thresholding to linear hyperplanes, and Bayesian treed linear models (BTLM- s) [8] employ linear hyperplanes for region-specific predictors. Lo- cal supervised learning through space partitioning (LSL-SP) [42] utilizes linear hyperplanes for both region-specific predictors and region specifiers. Although such models improve predictive accuracy over simple decision trees, their dense linear hyperplanes make the models difficult to understand. [44] studied a sparse treed model that aims to reduce the test-time cost. Eto et al. [11] proposed a variant of hierarchical mixture experts models that employs factorized asymptotic Bayesian inference for model selection (FAB/HMEs). Using the FAB framework [14], they enforce sparseness on region-specific linear predictors, which significantly improves interpretability over dense linear predictors, though their single-feature thresholding for region specifiers still restricts overall predictive ability. Supersparse linear integer models and their variants [26, 40] also learn highly sparse and interpretable model structures, which was also presented as a KDD 2014 Industrial and Government Track Invited Talk. A family of locally-linear models (fast local KSVMs [35], locally linear SVMs [25], clustered SVMs [15], and local deep kernel learning (LDKL) [23]) uses test-point-specific linear predictors. They do not have explicit regions but, rather, generate linear predictors on the fly. A major drawback of this approach for our purposes is that they can provide model information only with every single test point, which makes it difficult to understand overall prediction behaviors.

3. OT-SPAMS: OBLIQUE TREED SPARSE ADDITIVE MODELS

This section presents details of OT-SpAMs. We first describe the region specifiers and region-specific predictors for OT-SpAM and then derive the factorized asymptotic Bayesian inference in order to address the simultaneous model selection challenge.

3.1 OT-SpAMs

Our OT-SpAM is a variant of HMEs [22], which are tree structured probabilistic mixtures of experts models. In HMEs, region-specific predictors (leaf nodes in trees) are referred to as *experts*. Suppose we have observations $\{\mathbf{x}^n, y^n\}_{n=1}^N \sim \mathrm{X} \times \mathrm{Y}$, where $\mathrm{X} \in \mathrm{R}^D$ is the domain of covariates, $\mathrm{Y} \in \mathrm{R}$ (for regression tasks) or $\{0, 1\}$ (for classification tasks), N is the number of samples, and D is the data dimensionality. Each gate (non-leaf node) in the tree determines whether a data instance will go to its left or right branch. At the i-th gate ($i = 1, \ldots, G$, where G is the number of gates), let $z_i \in \{0, 1\}$ be the binary variable indicating which branch the instance $\mathbf{x}$ should go down (without loss of generality, let $z_i = 0$ represents the instance for going left). OT-SpAMs employ the following logistic hyperplane for their oblique region specifiers:

$$p(z_i|\mathbf{x}) = \left(\frac{1}{1+\exp(-\mathbf{w}^i \cdot \mathbf{x})}\right)^{1-z_i} \left(\frac{1}{1+\exp(\mathbf{w}^i \cdot \mathbf{x})}\right)^{z_i}, \tag{1}$$

where $\mathbf{w}^i$ is expected to be sparse for maintaining interpretability.

Let $\zeta^n = (\zeta^n_1, \ldots, \zeta^n_E) \in \{0, 1\}^E$ (E is the number of experts) represent an indicator of the expert to which $\mathbf{x}^n$ belongs, where $\zeta^n_j = 1$ stands for the instance of belonging to the j-th expert. Let G_i be the index set for the i-th gate, where G_i contains the indices of experts on the sub-tree of the i-th gate. Let E_j be the index set for the j-th expert, where E_j contains the indices of gates on the path from the root to the j-th expert. Given the region-specifier hyperplanes $\{\mathbf{w}^i\}_{i=1}^G$, the distribution on ζ^n can be described as follows:

$$p(\zeta^n|\mathbf{x}^n, \{\mathbf{w}^i\}_{i=1}^G) = \prod_{j=1}^E \prod_{i \in \mathrm{E}_j} \psi(\mathbf{x}^n, i, j)^{\zeta^n_j}, \tag{2}$$

where $\psi(\mathbf{x}^n, i, j)$ is the probability of $\mathbf{x}^n$'s going to the branch to which the j-th expert belongs at gate i, more specifically:

$$\psi(\mathbf{x}^n, i, j) = \begin{cases} p(z^n_i = 0|\mathbf{x}^n), & j \in \mathrm{G}^{\mathrm{left}}_i \\ p(z^n_i = 1|\mathbf{x}^n), & \text{otherwise} \end{cases}. \tag{3}$$

where $\mathrm{G}^{\mathrm{left}}_i$ is the index set of experts in the left sub-tree of the i-th gate.

Let us consider the following SAM:

$$f_j(\mathbf{x}) = \sum_{d=1}^D f_{jd}(\mathbf{x}_d), \tag{4}$$

where $f_{jd}(\cdot)$ is any smooth univariate function and many of them are expected to be zero (i.e., sparse). Notice that, if we set $f_{jd}(\mathbf{x}_d) = \theta_d \mathbf{x}_d$ with linear coefficients θ_d, then (4) will be reduced to a standard linear model. The generating distributions of y on the j-th expert is given by:

$$p(y|\mathbf{x}, \varphi_j) = \mathrm{N}(f_j(\mathbf{x}) - y, \sigma^2_j), \tag{5}$$

Table 1: Comparison of region specific predictive models (sp.=sparseness, s.f.=single feature, f.w.=feature-wise).

	SLM	SAM	DT	ODT	BTLM	FAB/HME	LSL-SP	**OT-SpAM**	LDKL
region	global		s.f. threshold	oblique	s.f. threshold		oblique		test-point specific
region sp.	X			×	X		×	X	N.A.
predictor	linear	f.w. nonlinear	constant		linear			f.w. nonlinear	linear
predictor sp.	X				×	X	×	X	×
ref.	[37]	[6, 7, 34]	[4]	[33, 39]	[8]	[11]	[42]	**this paper**	[23]

for regression, where $\varphi_j = (f_j, \sigma_j)$, and

$$p(y|\mathbf{x}, \varphi_j) = \frac{1}{1+\exp(f_j(\mathbf{x}))}^{1-y} \frac{\exp(f_j(\mathbf{x}))}{1+\exp(f_j(\mathbf{x}))}^{y}, \tag{6}$$

for classification, where $\varphi_j = f_j$.

In summary, the entire likelihood is given by:

$$p(\{y^n\}_{n=1}^N|\{\mathbf{x}^n\}_{n=1}^N, \{\varphi_j\}_{j=1}^E, \{\mathbf{w}^i\}_{i=1}^G) = \prod_{n=1}^N \sum_{j=1}^E p(y^n|\mathbf{x}^n, \varphi_j) p(\zeta_j^n|\mathbf{x}^n, \{\mathbf{w}^i\}_{i=1}^G). \tag{7}$$

3.2 Model Selection for OT-SpAM using FAB Framework

In order to learn OT-SpAMs, as well as parameter estimation, we have to address three model selection issues simultaneously:

M1: tree structure (the number of gates and experts, etc.).

M2: sparseness of region specifiers (logistic gates presented in (1)).

M3: sparseness of sparse additive experts.

To accomplish these model selection tasks, we employ FAB inference [14] for OT-SpAMs. Note that FAB has recently been used for learning treed sparse linear models [11], and this paper extends their framework to the learning of OT-SpAMs.

FAB inference maximizes the following Bayesian marginal log likelihood:

$$p(\{y^n\}_{n=1}^N|\{\mathbf{x}^n\}_{n=1}^N) = \max_q E_q \left[\log \frac{p(\{y^n\}_{n=1}^N, \{\zeta^n\}_{n=1}^N|\{\mathbf{x}^n\}_{n=1}^N)}{q(\{\zeta^n\}_{n=1}^N)}\right], \tag{8}$$

where q is an arbitrary distribution on $\{\zeta^n\}^N{}_{n=1}$ and the optimal q is $q^*(\{\zeta^n\}_{n=1}^N) = p(\{\zeta^n\}_{n=1}^N|\{y^n\}_{n=1}^N, \{\mathbf{x}^n\}^N{}_{n=1})$. Let Θ be $\Theta = [\mathbf{W}, \Phi]$ where $\mathbf{W} = [\mathbf{w}^1, ..., \mathbf{w}^G]$ and $\Phi = [\varphi_1, ..., \varphi_E]$. Laplace's method [43] is then applied to the numerator inside the log-function in (8) as follows:

$$p(\{y\}_{n=1}^N, \{\zeta\}_{n=1}^N|\{\mathbf{x}\}_{n=1}^N) \approx p(\{y\}^N{}_{n=1}, \{\zeta\}^N{}_{n=1}|\{\mathbf{x}\}^N{}_{n=1}, \bar{\Theta}) \prod_{i=1}^G \frac{(2\pi)^{\frac{D_{\mathbf{w}^i}}{2}}}{(\sum_{n=1}^N \sum_{j\in\mathbb{G}_i} \zeta_j^n)^{\frac{D_{\mathbf{w}^i}}{2}} |\bar{\mathrm{F}}_{\mathbf{w}^i}|^{1/2}} \prod_{j=1}^E \frac{(2\pi)^{\frac{D_{\varphi_j}}{2}}}{(\sum_{n=1}^N \zeta_j^n)^{\frac{D_{\varphi_j}}{2}} |\bar{\mathrm{F}}_{\varphi_j}|^{1/2}}, \tag{9}$$

where

$$\bar{\mathrm{F}}_{\mathbf{w}^i} = -\sum_{n=1}^N \sum_{j\in\mathbb{G}_i} \zeta_j^n \frac{\partial^2 \log p(\zeta^n|\mathbf{x}^n, \{\mathbf{w}^i\}_{i=1}^G)}{\partial \mathbf{w}^i \partial \mathbf{w}^{iT}}, \tag{10}$$

$$\bar{\mathrm{F}}_{\varphi_j} = -\sum_{n=1}^N \zeta_j^n \frac{\partial^2 \log p(y^n|\zeta^n, \mathbf{x}^n, \varphi_j)}{\partial \varphi_j \partial \varphi_j^T}. \tag{11}$$

$\bar{\Theta} = [\bar{\mathbf{W}}, \bar{\Phi}]$ is the maximum complete likelihood estimator and $D_\bullet$ denotes the dimensionality of $\bullet$.

Although Eto et al. [11] asymptotically ignore $|\bar{\mathrm{F}}_{\mathbf{w}^i}|^{1/2}$ and $|\bar{\mathrm{F}}_{\varphi_j}|^{1/2}$, using the law of large numbers, this paper considers the following upper bounds to obtain a better approximation, using Hadamard's inequality [30]:

$$|\bar{\mathrm{F}}_{\mathbf{w}^i}|^{1/2} \leq \left(\sum_{n=1}^N \sum_{j\in\mathbb{G}_i} \zeta_j^n\right)^{\frac{2}{D_{\mathbf{w}^i}}} \left(\sum_{n=1}^N \sum_{j\in\mathbb{G}_i} \frac{\zeta_j^n \partial^2 \log \psi(\mathbf{x}^n, i, j)}{\partial^2(\mathbf{w}^i \cdot \mathbf{x}^n)}\right)^{\frac{2}{D_{\mathbf{w}^i}}} \tag{12}$$

$$|\bar{\mathrm{F}}_{\varphi_j}|^{1/2} \leq \left(\sum_{n=1}^N \zeta_j^n\right)^{\frac{2}{D_{\varphi_j}}} \left(\sum_{n=1}^N \frac{\partial^2 \log p(y^n|x^n, \varphi_j)}{\partial^2 f_j}\right)^{\frac{2}{D_{\varphi_j}}}. \tag{13}$$

By substituting (9), (12) and (13) into (8), we obtain factorized information criterion (FIC) as follows:

$$\mathrm{FIC}(\{\mathbf{x}, y\}_{n=1}^N, \Theta) = \max_{q,\Theta} \left\{ \mathrm{L}(\{\mathbf{x}, y\}_{n=1}^N, \Theta, q) \right\}, \tag{14}$$

where

$$\begin{aligned}\mathrm{L}(\{\mathbf{x}, y\}_{n=1}^N, \Theta, q) = E_q \Big[&\log p(\{y\}_{n=1}^N, \{\zeta\}_{n=1}^N|\{\mathbf{x}\}_{n=1}^N, \Theta) \\ &- \sum_{i=1}^G \frac{||\mathbf{w}^i||_0}{2} \log\Big(\sum_{n=1}^N \sum_{j\in\mathbb{G}_i} \alpha_{ij}^n \zeta_j^n\Big) - \sum_{j=1}^E \frac{||f_j||_0}{2} \log\Big(\sum_{n=1}^N \eta_j^n \zeta_j^n\Big) \\ &- \sum_{n=1}^N \sum_{j=1}^E q(\zeta_j^n) \log q(\zeta_j^n)\Big],\end{aligned} \tag{15}$$

and

$$\alpha_{ij}^n = \frac{\exp(\mathbf{w}^i \cdot \mathbf{x}^n)}{(1+\exp(\mathbf{w}^i \cdot \mathbf{x}^n))^2} = \frac{\partial^2 \log \psi(\mathbf{x}^n, i, j)}{\partial^2(\mathbf{w}^i \cdot \mathbf{x}^n)}, \tag{16}$$

$$\eta_j^n = \begin{cases} \frac{1}{\sigma_j^2} & \text{for regression} \\ \frac{\exp f_j(\mathbf{x}^n)}{(1+\exp f_j(\mathbf{x}^n))^2} & \text{for classification} \end{cases}. \tag{17}$$

$||\mathbf{w}^j||_0$ and $||f_j||_0$ are the cardinalities of $\mathbf{w}_i$ and f_j, i.e, the number of non-zero $\mathbf{w}_{id}^i$ and f_{jd}, respectively. Here, for computational simplicity, we assume that the data is appropriately scaled in advance such that $\mathbf{x}^n \in [-1, 1]^D$.

Our new approximation, (12) and (13), results in a key difference from FIC for HMEs derived by Eto et al. [11], namely the regularization terms (wave underline) are adjusted with the factors α_{ij}^n and η_j^n (by setting $\alpha_{ij}^n = 1$, and $\eta_j^n = 1$, (15) becomes consistent with that of Eto et al. [11]). These factors come from the diagonal elements of $\mathrm{F}_{\mathbf{w}^i}$ and F_{φ_j}, which are empirical Fisher information matrices and provide natural measurements on the likelihood spaces [1]. It is worth noting that the previous FIC (i.e., $\alpha_{ij}^n = 1$ and $\eta_j^n = 1$) regularizes the model without relation to

Algorithm 1 FAB EM optimization for OT-SpAM

Input Data: $\{(x^n, y^n)\}_{n=1}^N$.
Input Parameters: D (maximum depth of the tree), δ (stopping condition), ε (shrinkage threshold).
Initialization: $t = 0$, $L^{(0)} = -\infty$, $\{\zeta^n\}_{n=1}^N \sim U[0, 1]$.
while $L^{(t)} - L^{(t-1)} > \delta$ **do**
  M-Step: Update $S_j^{(t)}$, $f_j^{(t)}$ and $\sigma_j^{(t)}$ (regression) using Algorithm 3.
  M-Step: Update $w_i^{(t)}$ using **Gate Optimization** as Algorithm 2.
  E-Step: Update $q^{(t+1)}(\zeta_j^n)$ using (18).
  Expert Shrinkage: Eliminate "non-effective" experts using (19).
  $t = t + 1$.
end while
Post-processing: Execute hard-gate post-processing (see [11] for details).

the metric space of $p(\{y\}_{n=1}^N, \{\zeta\}_{n=1}^N | \{x\}_{n=1}^N \Theta)$. On the other hand, our regularizers (wave underline) can naturally adjust the effect by taking the metric into account.

4. OPTIMIZATION ALGORITHM

To obtain the model which maximizes FIC (15), FAB employs EM-like alternating optimization on Θ (M-step) and q (E-step). The overall algorithmic framework is described in Algorithm 1. The superscription (t) represents the t-th EM iteration.

4.1 E-Step: updating variational distribution

From (15), we obtain the following update equation:

$$q^{(t)}(\zeta_j^n) \propto p(y^n | x^n, \varphi_j^{(t-1)}) \prod_{i \in E_j} \psi^{(t)}(x^n, i, j) \qquad (18)$$

$$\exp\left(-\frac{\|f_j\|_0 \eta_j^n}{2N_j}\right) \exp\left(\sum_{i \in E_j} -\frac{\|w^i\|_0 \alpha_{ij}^n}{2N_i}\right),$$

where $N_j = \sum_{n=1}^N \eta_j^n \zeta_j^n$ and $N_i = \sum_{n=1}^N \sum_{j \in G_i} \alpha_{ij}^n \zeta_j^n$. In contrast to standard EM algorithms, (18) has the additional terms marked with the waved underline. These terms come from the regularization terms in (15) (also marked with a waved underline). This causes a *shrinkage effect* [11, 14] through the EM iteration, i.e, more complex and smaller experts are penalized more, and we can safely eliminate "non-effective" experts from the model using a simple thresholding rule as follows:

$$\sum_{n=1}^N q^{(t)}(\zeta_j^n) < \delta. \qquad (19)$$

In practice, one could start from a sufficiently-large tree, after which the "shrinkage" scheme of OT-SpAM would find the proper size tree structure for capturing the data well. In this way, we have addressed the model selection issue **M1**.

4.2 M-Step: Learning Sparse Oblique Region Specifiers

We update the i-th gate by solving the following optimization problem (the w^i related terms in (15)):

$$w^{i(t)} = \arg\max_{w^i} \sum_{n=1}^N \sum_{j \in G_i} q^{(t-1)}(\zeta_j^n) \log \psi(x^n, i, j) - \frac{\|w^i\|_0}{2} \log\left(\sum_{n=1}^N \sum_{j \in G_i} \eta_j^n q^{(t-1)}(\zeta_j^n)\right)$$

Let G_{iL} be the index sets of experts on the left sub-tree of gate i, and G_{iR} be the index sets of experts on the right sub-tree of gate i. We can re-write the problem as follows:

$$w^{i(t)} = \arg\max_{w^i} Q(w^i) - \frac{\|w^i\|_0}{2} \log\left(\sum_{n=1}^N \sum_{j \in G_i} \eta_j^n q^{(t-1)}(\zeta_j^n)\right), \qquad (20)$$

where

$$Q(w^i) = \sum_{n=1}^N \sum_{j \in G_i} q(\zeta_j^n) \left[\frac{\sum_{j \in G_{iL}} q(\zeta_j^n)}{\sum_{j \in G_i} q(\zeta_j^n)} \log \frac{\exp(w^i \cdot x^n)}{1 + \exp(w^i \cdot x^n)} + \frac{\sum_{j \in G_{iR}} q(\zeta_j^n)}{\sum_{j \in G_i} q(\zeta_j^n)} \log \frac{1}{1 + \exp(w^i \cdot x^n)}\right]. \qquad (21)$$

This problem can be seen as a sparsity-regularized generalized logistic regression problem: i) unlike the standard regression, here the response is any number in $[0, 1]$ ($\frac{\sum_{j \in G_{iL}} q(\zeta_j^n)}{\sum_{j \in G_i} q(\zeta_j^n)}$ and $\frac{\sum_{j \in G_{iR}} q(\zeta_j^n)}{\sum_{j \in G_i} q(\zeta_j^n)}$ in this problem) and ii) there is a weight for each instance: $\sum_{j \in G_i} q(\zeta_j^n)$.

Problem (20) is non-convex (due to the L_0 regularization), and we have adopted a greedy strategy [38] to get an approximate solution. Details are shown in Algorithm 2. Let $S \subseteq [D]$ be the set of selected features. Also, we denote the maximizer of Q as follows:

$$\hat{w}^i(S) = \max_{w^i(S)} Q(w^i(S)), \qquad (22)$$

where solving (22) is a constrained, weighted logistic regression problem. At each iteration, we selected the feature that maximizes the gradient absolute value $|\nabla_d Q(w^i)|$, which is

$$|\nabla_d Q(w^i)| = \left|\sum_{j \in G_i} q(\zeta_j^{1:N}) \circ R^{1:N}) \cdot x_d^{1:N}\right|, \qquad (23)$$

where $\circ$ is the Hadamard product, and

$$R^{1:N} = \frac{\sum_{j \in G_{iL}} q(\zeta_j^{1:N})}{\sum_{j \in G_i} q(\zeta_j^{1:N})} - \frac{1}{1 + \exp(-w^{i(k)} \cdot x^{1:N})}$$

and then solved the constrained weighted logistic regression problem, until

$$Q(w^{i(k)}) - Q(w^{i(k-1)}) \le \log\left(\sum_{n=1}^N \sum_{j \in G_i} \eta_j^n q^{(t-1)}(\zeta_j^n)\right), \qquad (24)$$

was satisfied, where (k) is the iteration index in Algorithm 2. In this way, we have addressed the model selection issue **M2**.

4.3 M-Step: Learning Local Experts

In order to optimize the j-th local expert f_j, we introduce the following model:

$$f_{jd}(x) = \sum_{m=1}^M \beta_{jd}^m g_m(x), \qquad (25)$$

Algorithm 2 Gate Optimization for OT-SpAM

1: **for** $i = 1, \ldots, G$ **do**
2: **Initialization** $S_i^{(k)} = \emptyset$, $k = 0$, $\mathrm{w}^{i(k)} = 0$, $\mu_i^{1:N(k)} = 1/(1 + \exp(\mathrm{w}^{i(k)} \cdot \mathrm{x}^{1:N})$.
3: **while TRUE do**
4: Select feature $d^{(k)} = \arg\max_{d \in S_i^{(k)}} |\nabla_d Q(\mathrm{w}^i)|$ according to (23).
5: $k = k + 1$.
6: Update $S_i^{(k)} = S_i^{(k)} \cup d^{(k)}$, $\mathrm{w}^{i(k)} = \hat{\mathrm{w}}(S_i^{(k)})$, $\mu_i^{1:N(k)} = 1/(1 + \exp(-\mathrm{w}^{i(k)} \cdot \mathrm{x}^{1:N})$.
7: **if** (24) is satisfied **then**
8: $k = k - 1$, and **Break**.
9: **end if**
10: **end while**
11: Output $\mathrm{w}^{i(k)}$.
12: **end for**

Algorithm 3 Greedy Additive Regression for OT-SpAM

1: **Input** Data: $\{(\mathrm{x}^n, y^n)\}_{n=1}^N$, $q^{(t)}(\zeta_j^n)$, $\sigma_j^{(t-1)}$.
2: **for** $j = 1, \ldots, E$ **do**
3: **Initialization** $S_j^{(k)} = \emptyset$, $k = 0$, $\alpha = \sum_{n=1}^N y^n/N$, $\hat{f}_j^{(k)} = \alpha$, Residual $R^{(k)} = y^{1:N} - \hat{f}_j^{(k)}(\mathrm{x}^{1:N})$.
4: **while TRUE do**
5: $k = k + 1$.
6: Select feature $d^{(k)}$ using (27).
7: Fit $\hat{f}_{jd}^{(k)}$ (by updating $\beta_{jd}^{1:M}$) using (28)
8: Update $S_j^{(k)} = S_j^{(k)} \cup d^{(k)}$, $\hat{f}_j^{(k)} = \alpha + \sum_{d \in S_j^{(k)}} \hat{f}_{jd}(\mathrm{x}_j)$.
9: Update residual $R^{(k)}$ using (29) for regression and (30) for classification.
10: **if** (32) is satisfied **then**
11: $k = k - 1$, and **Break**.
12: **end if**
13: **end while**
14: Update $\hat{f}_j^{(t)} = \hat{f}_j^{(k)}$, $\sigma_j^{(t)} = \|R^{(k)}\|_2$.
15: **end for**

where g_m is a pre-defined smooth basis function and M is the number of basis functions (in our experiments, we use P-spline functions as g_m). Here our parameterization is changed to $\varphi_j = \beta_j$, where $\beta_j = (\beta_{j1}^{1:M}, \ldots, \beta_{jD}^{1:M})$. We then update the j-th local expert by solving the following optimization problem:

$$\beta_j^{(t)} = \arg\max_{\beta_0} \sum_{n=1}^N \sum_{j=1}^D q^{(t-1)}(\zeta_j^n) \log p(y^n | \mathrm{x}^n, \beta_j, \sigma_j^{2(t-1)}) - \sum_{j=1}^D \frac{\|\beta_j\|_{\infty,0}}{2} \log\Big(\sum_{n=1}^N \beta_j^n \zeta_j^n\Big), \quad (26)$$

where $\|\beta_j\|_{\infty,0} = \|\|\beta_{j1}^{1:M}\|_\infty, \|\beta_{j2}^{1:M}\|_\infty, \ldots, \|\beta_{jD}^{1:M}\|_\infty\|_0$. Notice that we can simply ignore $\sigma_j^{2(t-1)}$ when we consider the classification case.

Problem (26) is reduced to the optimization of weighted GLM under group sparsity regularization. This paper adopts the greedy optimization method summarized in Algorithm 3. Note that existing works on greedy group selection [28] (or additive forward regression [27]) include proposals for addressing the greedy group feature selection problem. In contrast to these, Algorithm 3 has the following differences: i) at the feature selection stage, [27] s- elects the feature that maximizes the alignment between residuals and fitted responses. Here we directly select the feature with the maximum gradient norm:

$$d^{(k)} = \arg\max_{d \notin S_j^{(k)}} \|g^{1:M}(\mathrm{x}_d^{1:N})(R^{(k)} \circ q^{(t)}(\zeta_j^{1:N}))\|_2. \quad (27)$$

This gradient criterion avoids having to fit the model $O(D)$ times, which makes the selection process much faster; ii) [27, 28] use an orthogonal matching pursuit type fitting procedure [38] (that is, after selecting one new feature, the model is re-fitted using the newly selected feature pool). Rather than this approach, we use a matching pursuit [29] type method to speed up the algorithms, i.e., we just add the new fitted univariate function without re-fitting the model. The fitting equation (derived by solving a weighted least squares) is described as follows :

$$\beta_{jd}^{1:M} = (\mathrm{G}^T \mathrm{H} \mathrm{G})^{-1} \mathrm{G}^T R^{(k)}. \quad (28)$$

where $\mathrm{G} \in \mathrm{R}^{N \times M}$ is the feature matrix such that $\mathrm{G}_{nm} = g_m(\mathrm{x}_d^n)$ and $\mathrm{H} \in \mathrm{R}^{N \times N}$ is the diagonal weighting matrix such that $\mathrm{H}_{nn} = q^{(t)}(\zeta_j^n)^2$. These special designs make the algorithm much faster than the procedures in [27, 28] by avoiding repeated model fitting and re-fitting. Though it might be less accurate in feature selection and model fitting (but not too much when the basis functions are not highly correlated with each other), the hard-gate post-processing proposed in [11] (the step 11 of Algorithm 1) makes the final model more stable and reliable, as we will see in Section 5.

The difference between classification and regression is in the update of the residual. For regression, we can naturally define the residual as follows:

$$R^{(k)} = y^{1:N} - \hat{f}_j^{(k)}(\mathrm{x}^{1:N}) \quad (29)$$

The residual of logistic loss for classification is not so obvious, but we follow [17], which defines it in terms of the updating direction of the Newton step as follows:

$$R^{(k)} = \frac{y^{1:N} - \mu_j^{1:N(k)}}{\mu_j^{1:N(k)} \circ (1 - \mu_j^{1:N(k)})}, \quad (30)$$

where

$$\mu_j^{1:N(k)} = \frac{\exp(\hat{f}_j^{(k)}(\mathrm{x}^{1:N}))}{1 + \exp(\hat{f}_j^{(k)}(\mathrm{x}^{1:N}))}. \quad (31)$$

The stopping condition is defined as follows:

$$\sum_{n=1}^N q^{(t-1)}(\zeta_j^n) \log p(y^n | \mathrm{x}^n, \hat{\beta}_j^{(k)}, \sigma_j^{(t-1)}) - \log p(y^n | \mathrm{x}^n, \hat{f}_j^{(k-1)}, \sigma_j^{(t-1)}) \leq \log\Big(\sum_{n=1}^N \eta_j^n \zeta_j^n\Big). \quad (32)$$

In this way, we have addressed the model selection issue **M3**.

5. SIMULATION STUDY: MODEL SELECTION AND VISUALIZATION

This section presents results of simulation studies and demonstrates our FAB-based model selection for OT-SpAMs. In order to make OT-SpAMs interpretable, we proposes a visualization method that employs individual local sparse additive experts.

5.1 Simulation Setup

We generated $N = 5000$ data points in which each instance is described by $D = 15$ features, and the features are uniformly distributed in [0,1], i.e. $X \sim U[0, 1]^D$. The true tree structure is

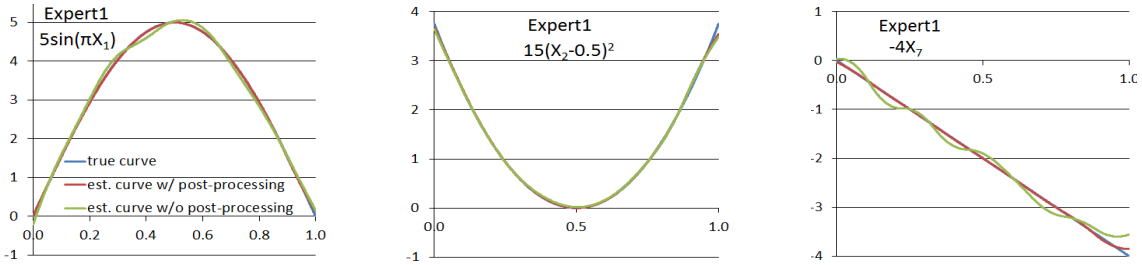

Figure 1: Estimated additive functions for Expert1 in the simulations. The horizontal and vertical axes represent, respectively, the original feature and the estimated sparse additive feature.

shown in (A). It has 4 experts, each of which uses 2 − 3 features, and the partition nodes use linear functions of 2 features.

$$2.5X_9 - 1.6X_{10} < 0.613$$

$$1.5X_6 - 0.7\overline{X_4 < 0.358} \quad 1.6X_3 - 0.5X_{12} < 0.715$$

Expert 1 Expert 2 Expert 3 Expert 4

(A) True tree model

We generated response Y of the instances, in accord with the experts they belongs to, in the following way:

- Expert 1: $Y = 5\sin(\pi X_1) + 15(X_2 - 0.5)^2 - 4X_7 + 1 + \varepsilon$, $o \sim 0.01\mathsf{N}(0, 1)$.
- Expert 2: $Y = 7/(1 + \exp(5 - 10X_5)) + 5\sin(\pi X_7^2) - 5X_8 + 2 + \varepsilon$, $\varepsilon \sim 0.01\mathsf{N}(0, 1)$.
- Expert 3: $Y = 8|X_1 - 0.5| + 7(3X_8 - 2)^2 - 4 + \varepsilon$, $\varepsilon \sim 0.01\mathsf{N}(0, 1)$.
- Expert 4: $Y = 5\cos(2\pi X_2) + 5X_8 + 2\log(100X_{10} + 3) - 3 + \varepsilon$, $\varepsilon \sim 0.01\mathsf{N}(0, 1)$.

In this simulation, we set the initial tree-depth to $\mathsf{D} = 4$ (i.e., the initial number of experts was 16), the shrinkage threshold to $\varrho = 0.06N$, and the stopping threshold to $\delta = 10^{-5}$. Also, since oblique region specifiers using many features are hard to interpret, we set the maximum number of features used in each partition node to 3. Additionally, we employed P-spline functions (a family of B-splines with a smoothness penalty [10]) as g_m in (25). We chose the penalty parameter for P-splines as 0.5, the spline degree as 3, the number of knots as 6.

5.2 Model Selection Results

The estimated tree structure is shown in (B). There were 16 experts at the start, irrelevant experts were gradually pruned from the model by means of FAB regularization, and, at the convergence point, our method almost completely recovered[1] the true tree structures with exactly the same features in each gate (oblique hyperplane).

[1]The partition functions in (B) have been properly scaled for comparison with the functions in (A), since scaling the functions does not change the decision boundary.

$$2.5X_9 - 1.486X_{10} < 0.663$$

$$1.5X_6 - 0.683\overline{X_4 < 0.369} \quad 1.6X_3 - 0.506X_{12} < 0.698$$

Expert 1 Expert 2 Expert 3 Expert 4

(B) Estimated tree model

Figure 1 shows the estimated additive functions of Expert1. Since we employ a matching pursuit type of optimization in the M-step, there is marginal estimation error in the estimated fea- ture functions before post-processing (green curves). After post-processing, the true (blue curves) and estimated curves (red curves) are quite consistent. Although we omit results for the other experts, we obtained similarly good estimation results for those as well.

These results empirically demonstrate strong model selection capability in addressing **M1**, **M2** and **M3** simultaneously.

5.3 Visualization of Local Sparse Additive Experts

Since each sparse additive feature f_{jd} is (feature-wise) nonlinear, visualization is critically important to maintain model interpretability. This paper proposes a stacked area plot to visualize sparse additive features. Figure 2 shows the visualization for Expert1 (we shifted Expert1 to the negative side in this figure to more easily explain our visualization method). The left-hand figure is a simple line plot of estimated feature functions w.r.t. $X1$, $X2$, and $X7$. The line plot would be difficult to see if several features were selected and overlapped one another in a single plot. To avoid this, we employ a stacked area plot that is constructed as follows. First, individual feature functions are separated into positive and negative sides, as shown in the middle column of Figure 2. The s- tacked area plot was then built by combining positive and negative stacked area plots (the right-hand figure). As is shown, we are able to avoid "ugly" overlapping and can clearly see how each input fea- ture "nonlinearly" contributes to the target signal. We visualize the stacked area plot for each expert, and the combination of standard tree visualization with the stacked area plots provides a full picture of nonlinear model behaviors.

6. BENCHMARK EVALUATION OF PREDICTIVE ACCURACY

We evaluated OT-SpAMs on 24 public benchmark data sets, available from the UCI Machine Learning Repository [2], for both regression and classification tasks. Table 2 summarizes the statistics for these data sets. We used the same initial tree-depth and ter-

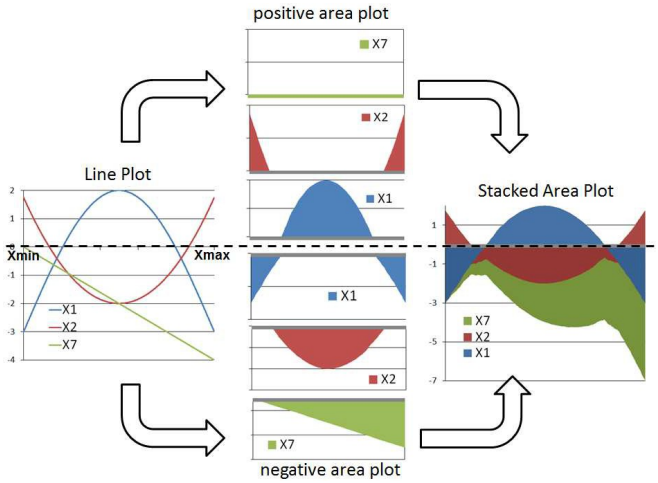

Figure 2: Stacked area visualization for the learned Expert1 in the simulation study. The horizontal and vertical axes represent, respectively, the original feature and the estimated sparse additive feature.

Table 2: List of benchmark datasets.

ID	Name	#Instances	#Features	Task
D1	Auto-mpg	398	4	Regression
D2	Boston-housing	506	13	Regression
D3	Stock	950	9	Regression
D4	Space-ga	3107	6	Regression
D5	Abalone	4177	8	Regression
D6	ParkinsonM	5875	20	Regression
D7	Cpusmall	8192	12	Regression
D8	Kinematics	8192	8	Regression
D9	Puma8nh	8192	8	Regression
D10	Comp-acti	8192	21	Regression
D11	Ailerons	13750	40	Regression
D12	Cadata	20640	8	Regression
D13	Banana	400	2	Classification
D14	Australian	690	14	Classification
D15	Pima Diabetes	768	8	Classification
D16	Fourclass	862	2	Classification
D17	Splice	1000	60	Classification
D18	Banknote	1372	4	Classification
D19	Titanic	2201	3	Classification
D20	Svmguide1	7089	4	Classification
D21	EEG-eyestate	14980	14	Classification
D22	Magic04	19120	10	Classification
D23	Cod-rna	59535	8	Classification
D24	Ijcnn1	141691	22	Classification

mination conditions as in the simulations and evaluated root mean squared error (RMSE) for regression and accuracy in classification.

For regression tasks, we compared OT-SpAM with the following methods: OLS (ordinary least squares using the full set of the features), RegTree[2] (a classical regression tree model [4]), FAB/HME [11], AM (additive models [16] using the full set of the features), and SVR-RBF [36]. For classification tasks, we compared OT-SpAM with the following methods: LLR (linear logistic regression using the full set of the features), CART[3] [4], LSL-SP[4] [42], FAB/HME (we adopted the method described in [11] with logistic model), LDKL [23], ALR (additive logistic regression [16] using the full set of the features), DLR[5] [6], and SVM-RBF[6] [9]. The parameters in SVR-RBF, LSL-SP, LDKL, DLR, and SVM-RBF were optimized on the basis of 10-fold cross validation on training data. Note that we used all features for linear and additive models (OLS, AM, LLR and ALR). The primary focus here was on accuracy evaluation, and they performed better with all features (without sparse regularization).

Table 3 and Table 4 report the 10-fold averaged cross validation RMSE and classification accuracy, respectively. From these results, we have the following observations:

- For regression tasks, OT-SpAMs achieved the lowest RMSE values in most cases. Both AMs and FAB/HMEs also performed much better than OLS and RegTrees. OT-SpAMs took advantages of both methods and performed competitively with SVR-RBF (or sometimes even outperformed it).

- For classification tasks, similar observations were obtained, i.e., FAB/HMEs and ALRs performed better than LLRs, and OT-SpAMs usually outperformed both them and state-of-the-art additive models (ALRs and DLRs). LDKL also performed well, but it is worth noting that LDKL produces a predictor at every single data point and that no interpretation of regions is provided, as may be seen in Table 1. We observed that OT-SpAMs performed slightly worse than SVM-RBFs and sacrificed accuracy for interpretability, though, except for D21, the sacrifice was not significant.

- On these data sets, OT-SpAM usually output treed models with 5-8 experts, and these models were reasonably interpretable. OT-SpAM selected different fractions of features, depending on the data sets used.

In summary, we conclude that OT-SpAMs sacrificed minimum accuracy loss for interpretability, w.r.t. fully non-parametric methods, by maintaining interpretable treed region structures and feature-wise sparse nonlinear expert structures.

7. REAL WORLD APPLICATION: SALES FORECASTING

In the retail industry, sales forecasting is a key component of advanced store management. Let us consider three scenarios:

A) store inventory management requires forecasting every 6 hours for 2-day to 1-week periods. Accurate forecasting reduces disposal loss, and model interpretability lets store managers safely use a forecasting-based ordering system.

B) store assortment planning requires forecasting every 1 day for 1-week to 3-week periods. Accurate forecasting increases revenue w.r.t. shelf-space, and model interpretability helps marketers to hypothesize good assortment strategies.

C) production planning requires forecasting every 1 week for 2 month periods. Accurate forecasting reduces supply-chain inventory losses, and model interpretability helps marketers to plan release timing for new products.

We applied OT-SpAM to sales forecasting of sweet bakery products in a middle-size supermarket located in a residential area of

[2]We use the built-in `RegressionTree` class in MATLAB
[3]We use the built-in `ClassificationTree` class in MATLAB
[4]http://blogs.bu.edu/joewang/code/
[5]http://www.cse.wustl.edu/~wenlinchen/project/DLR/
[6]For SVR-RBF and SVM-RBF, we use the LIBSVM package [5].

Table 3: Comparison of test RMSE values on benchmark datasets. The best and second best methods (except SVR-RBF) are highlighted in bold and *bold italic* faces, respectively.

ID	OLS	RegTree	FAB/HME	AM	OT-SpAM	SVR-RBF
D1	5.10 ± 0.53	6.67 ± 1.33	3.29 ± 0.32	***2.88 ± 0.43***	**2.79 ± 0.52**	3.13 ± 0.48
D2	5.38 ± 0.86	9.06 ± 2.92	3.72 ± 0.96	***3.65 ± 0.72***	**3.41 ± 0.55**	5.65 ± 0.80
D3	2.32 ± 0.09	1.99 ± 0.34	2.28 ± 0.39	***1.33 ± 0.11***	**1.02 ± 0.10**	0.91 ± 0.09
D4	0.14 ± 0.01	0.28 ± 0.03	***0.13 ± 0.01***	***0.13 ± 0.02***	**0.12 ± 0.02**	0.10 ± 0.01
D5	2.32 ± 0.17	5.57 ± 0.22	***2.27 ± 0.15***	2.31 ± 0.10	**2.22 ± 0.12**	2.17 ± 0.11
D6	7.30 ± 0.10	**0.90 ± 0.34**	4.96 ± 0.57	5.11 ± 0.06	***2.99 ± 0.19***	2.95 ± 0.07
D7	16.2 ± 0.31	7.57 ± 0.21	5.14 ± 0.62	***3.79 ± 0.22***	**3.26 ± 0.35**	4.56 ± 0.49
D8	0.29 ± 0.00	0.40 ± 0.01	0.24 ± 0.01	***0.20 ± 0.00***	**0.19 ± 0.00**	0.07 ± 0.01
D9	4.67 ± 0.09	8.44 ± 0.31	***4.18 ± 0.21***	4.24 ± 0.10	**3.31 ± 0.08**	3.34 ± 0.09
D10	15.5 ± 0.30	6.73 ± 0.28	5.12 ± 0.40	***3.57 ± 0.18***	**2.79 ± 0.61**	5.07 ± 0.35
D11	(2.50 ± 2.2) $\cdot 10^{-4}$	(4.27 ± 0.1) $\cdot 10^{-4}$	(***1.70 ± 0.0***) $\cdot 10^{-4}$	(1.73 ± 0.0) $\cdot 10^{-4}$	(**1.66 ± 0.0**) $\cdot 10^{-4}$	(5.97 ± 1.2) $\cdot 10^{-4}$
D12	(7.48 ± 0.1) $\cdot 10^{4}$	(12.6 ± 0.3) $\cdot 10^{4}$	(6.82 ± 0.1) $\cdot 10^{4}$	(***6.48 ± 0.1***) $\cdot 10^{4}$	(**5.97 ± 0.1**) $\cdot 10^{4}$	(11.8 ± 0.0) $\cdot 10^{4}$

Table 4: Comparison of test classification accuracy on benchmark datasets. The best and second best methods (except SVM-RBF) are highlighted in bold and *bold italic* faces, respectively.

ID	LLR	CART	LSL-SP	FAB/HME	LDKL	ALR	DLR	OT-SpAM	SVM-RBF
D13	68.5 ± 4.4	***84.7 ± 6.4***	69.7 ± 7.1	76.7 ± 8.5	**88.7 ± 4.6**	76.2 ± 8.1	67.0 ± 7.1	82.2 ± 7.5	91.3 ± 4.0
D14	84.6 ± 3.5	86.2 ± 4.2	86.2 ± 3.7	85.3 ± 3.7	85.6 ± 4.1	***86.5 ± 2.4***	***86.5 ± 3.5***	**87.1 ± 3.9**	85.5 ± 3.7
D15	69.6 ± 4.7	73.9 ± 8.0	74.9 ± 6.4	69.7 ± 8.3	75.7 ± 7.9	***76.5 ± 7.5***	***76.5 ± 6.7***	**77.5 ± 7.1**	75.8 ± 8.8
D16	75.0 ± 6.1	95.8 ± 2.2	78.6 ± 6.2	76.5 ± 6.0	**96.8 ± 3.3**	90.0 ± 3.4	75.8 ± 4.7	***96.1 ± 1.8***	99.8 ± 0.5
D17	80.4 ± 3.3	90.7 ± 3.6	83.0 ± 5.8	79.2 ± 3.3	85.4 ± 1.1	90.7 ± 2.2	***90.8 ± 2.0***	**92.5 ± 2.2**	86.1 ± 3.0
D18	86.0 ± 2.5	97.4 ± 1.5	98.1 ± 1.6	93.4 ± 1.9	**99.9 ± 0.2**	98.9 ± 0.9	88.8 ± 1.4	***99.1 ± 0.7***	100 ± 0.0
D19	77.1 ± 0.4	**79.1 ± 0.7**	78.1 ± 0.4	77.6 ± 0.5	***79.0 ± 0.7***	77.8 ± 0.3	77.6 ± 0.3	***78.3 ± 0.5***	78.5 ± 0.0
D20	89.1 ± 0.5	96.7 ± 0.8	93.7 ± 0.7	93.8 ± 0.9	96.2 ± 1.1	***96.8 ± 0.7***	95.5 ± 1.0	**97.1 ± 0.6**	96.9 ± 0.1
D21	58.0 ± 0.6	**82.7 ± 0.7**	***76.5 ± 1.2***	60.8 ± 1.4	55.1 ± 0.0	57.9 ± 1.8	55.1 ± 0.1	59.7 ± 2.3	81.6 ± 0.1
D22	78.9 ± 1.0	83.8 ± 0.8	81.4 ± 1.3	81.7 ± 1.4	***85.0 ± 0.8***	84.9 ± 0.8	78.6 ± 0.9	**85.8 ± 1.1**	87.2 ± 0.1
D23	89.6 ± 0.4	93.8 ± 0.3	91.1 ± 0.3	91.0 ± 0.5	***94.0 ± 1.2***	***94.0 ± 0.2***	77.6 ± 0.4	**94.2 ± 0.2**	95.6 ± 0.0
D24	92.1 ± 0.5	**97.4 ± 0.4**	90.4 ± 0.3	93.9 ± 2.2	94.2 ± 1.7	93.7 ± 0.6	91.1 ± 0.8	***95.6 ± 1.9***	98.6 ± 0.1

Tokyo[7]. Our primary target scenarios were A) and B), and we set the target variable to total sales of sweet bakery products for one day one week later. We used three years of historical data whose time resolution was daily. The first two years (731 sam- ples) were used for training, and the other year (365 samples) was used for testing. Table 5 summarizes variables used for the fore- casting. There are 30 input features in total. In addition to sales information (x20, x21 and the target variable), we independently collected weather related variables (x2-x19) and also added calen- dar information (x22-x30). All numerical variables, including the target variable, were standardized in advance. Experimental set- tings were the same as those of Sections 5 and 6, except the initial tree-depth (here we use $D = 5$). Figure 3 shows the forecasting results for the test period. As can be seen, OT-SpAM achieved fairly good forecasting.

$$-X_{17} + 0.55X_{20} \leq -0.97$$

$X_{20} \leq 1.1$ $\qquad$ $X_{29} \leq 0.49$

Expert 1 $\quad -X_{21} + 0.11X_{29} \leq -0.7$ $\qquad$ Expert 4 $\quad$ Expert 5

Expert 2 $\quad$ Expert 3

(C) Estimated tree model for sales forecasting

[7]The data has been provided by KDP-SP Co., LTD, http://www.ksp-sp.com.

The estimated tree structure is shown in (C). The region-specifier employed average pressure (x17), sales histories (x20 and x21), and weekday flag (x29). Taking into account the fact that average pressure in Tokyo is relatively high during May to September, the region-specifier identified the following clusters:

- Expert1: in-season (high average sales) during early summer to autumn.
- Expert2: off-season (low average sales) during early summer to autumn.
- Expert3: other season (middle average sales) during early summer to autumn.
- Expert4: weekday during autumn to early summer.
- Expert5: holiday during autumn to early summer.

Figure 4 provides our stacked area plot for individual experts. We can characterize the experts as follows:

- Expert1: Products in this category (sweets bakery) are sold a lot on Friday. The largest bias value among the experts supports our hypothesis that this cluster corresponds to in-season.
- Expert2: The small responses (i.e., the scale of the vertical axis is small) supports the hypothesis that this cluster corresponds to off-season, and the sales are small without relation to weather.
- Expert3: The response for daylight (x13) is high in the middle area of the horizontal axis. Since mid-summer daylight hours are long in Tokyo, this result indicates that sunny days tend to have large sales.
- Expert4: We can observe a strong response w.r.t the sales of 1 week previous but the peak is somehow shifted to the left-hand side. This might indicate a natural decrease in sales following a promotional campaign.

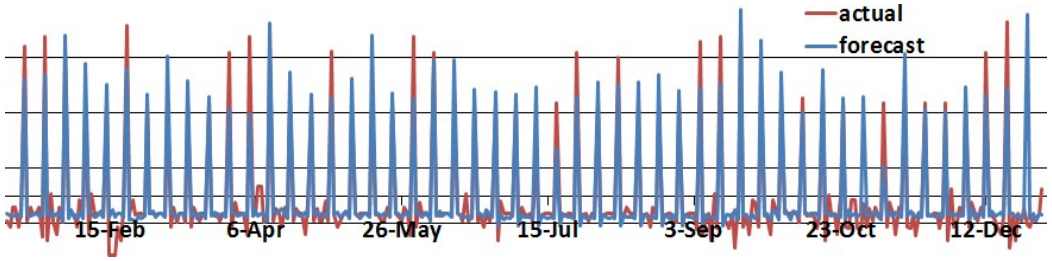

Figure 3: Sales forecasting results for the test period (one year) using OT-SpAMs.

Table 5: List of variables in the sales forecasting dataset. N and B stand for numerical and binary values. weather1 and weather2 are forecasting (1 week ahead) and history (1 week ago), respectively.

ID	name	value	type
x1	dates from Jan. 1st.	N	date
x2	rainfall forecast (mm)	N	weather1
x3	ave. temp. forecast (degree)	N	weather1
x4	daylight forecast (hours)	N	weather1
x5	snowfall forecast (cm)	N	weather1
x6	humidity forecast (%)	N	weather1
x7	cloudiness forecast (10%)	N	weather1
x8	ave. pressure forecast (hPa)	N	weather1
x9	max temp. forecast (degree)	N	weather1
x10	min temp. forecast (degree)	N	weather1
x11	rainfall history (mm)	N	weather2
x12	average temp. history (degree)	N	weather2
x13	daylight history (hours)	N	weather2
x14	snowfall history (cm)	N	weather2
x15	humidity history (%)	N	weather2
x16	cloudiness history (10%)	N	weather2
x17	ave. pressure history (hPa)	N	weather2
x18	max temp. history (degree)	N	weather2
x19	min temp. history (degree)	N	weather2
x20	sales (1 week ago)	N	sales history
x21	sales (2 weeks ago)	N	sales history
x22	Sunday	B	calendar information
x23	Monday	B	calendar information
x24	Tuesday	B	calendar information
x25	Wednesday	B	calendar information
x26	Thursday	B	calendar information
x27	Friday	B	calendar information
x28	Saturday	B	calendar information
x29	Weekday	B	calendar information
x30	Holiday	B	calendar information

- Expert5: The sales are low on Saturdays.

The above observations can be transformed into insights for improve store operations, such as:

A) store inventory management: In order to avoid excessive inventory, store managers should take into account the strong possibility of a post-promotional-campaign slump in sales. Further, store managers should increase the number of displayed items of this product category, as this may increase store revenue.

B) store assortment planning: store managers should consider possible adjustments to the product line-up in this product category since the sales trends may change.

These insights are still hypotheses and must be evaluated in real stores, but we believe that the above results demonstrate high interpretability of OT-SpAMs in the real world applications.

8. SUMMARY AND FUTURE WORK

We have proposed oblique treed sparse additive models, novel extensions of generalized additive models for heterogeneous data analysis that employs the learning of hierarchical mixtures of sparse additive models. We have presented a Bayesian learning algorithm which fully automates space partitioning and feature selection, making the proposed approach nearly parameter free. Promising empirical results have been obtained for both simulated and real-world data. Future work will address the theoretical understanding and computational efficiency of OT-SpAMs, as well as extensions to such more general data mining problems as multi-class classification and Poisson regression.

9. ACKOWLEDGEMENTS

The majority of the work was done during the internship of the first author at NEC Laboratories America, Cupertino, CA.

References

[1] S. Amari and H. Nagaoka. *Methods of Information Geometry*, volume 191 of *Translations of Mathematical monographs*. Oxford University Press, 2000.

[2] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[3] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[4] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.

[5] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011.

[6] W. Chen, Y. Chen, Y. Mao, and B. Guo. Density-based logistic regression. In *KDD*, pages 140–148, 2013.

[7] W. Chen, Y. Chen, and K. Q. Weinberger. Fast flux discriminant for large-scale sparse nonlinear classification. In *KDD*, pages 621–630, 2014.

[8] H. A. Chipman, E. I. George, and R. E. Mcculloch. Bayesian treed generalized linear models. *Bayesian Statistics*, 7, 2003.

[9] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[10] P. H. C. Eilers and B. D. Marx. Flexible smoothing with b-splines and penalties. *Statistical Science*, 11(2):89–121, 05 1996.

[11] R. Eto, R. Fujimaki, S. Morinaga, and H. Tamano.

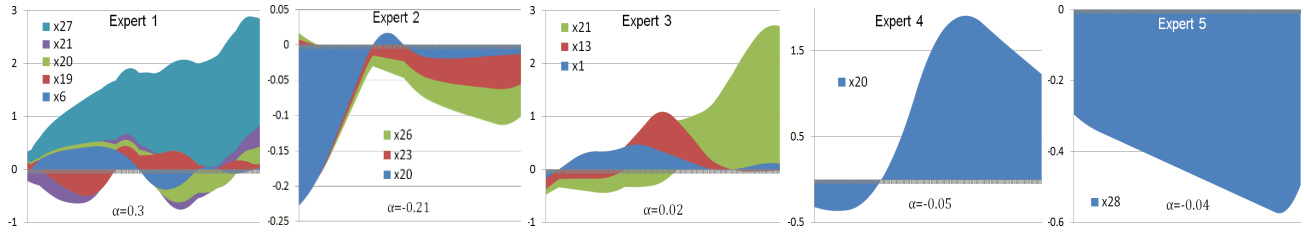

Figure 4: Estimated additive functions for sales forecasting data.

Fully-automatic bayesian piecewise sparse linear models. In *AISTATS*, pages 238–246, 2014.

[12] A. A. Freitas. Comprehensible classification models: A position paper. *SIGKDD Explor. Newsl.*, 15(1):1–10, Mar. 2014.

[13] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[14] R. Fujimaki and S. Morinaga. Factorized asymptotic bayesian inference for mixture modeling. In *AISTATS*, pages 400–408, 2012.

[15] Q. Gu and J. Han. Clustered support vector machines. In *AISTATS*, pages 307–315, 2013.

[16] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer-Verlag, 2001.

[17] T. J. Hastie and R. J. Tibshirani. *Generalized additive models*. London: Chapman & Hall, 1990.

[18] B. Hayete and J. R. Bienkowska. Gotrees: predicting go associations from protein domain composition using decision trees. *Pacific Symposium on Biocomputing*, pages 127–138, 2005.

[19] G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[20] J. Huang, J. L. Horowitz, and F. Wei. Variable selection in nonparametric additive models. *Ann. Statist.*, 38(4):2282–2313, 08 2010.

[21] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decis. Support Syst.*, 51(1):141–154, Apr. 2011.

[22] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6(2):181–214, 1994.

[23] C. Jose, P. Goyal, P. Aggrwal, and M. Varma. Local deep kernel learning for efficient non-linear SVM prediction. In *ICML*, pages 486–494, 2013.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012.

[25] L. Ladicky and P. H. S. Torr. Locally linear support vector machines. In *ICML*, pages 985–992, 2011.

[26] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan. An interpretable model for stroke prediction using rules and bayesian analysis. In *KDD Workshop on Data Science for Social Good*, 2014.

[27] H. Liu and X. Chen. Nonparametric greedy algorithms for the sparse learning problem. In *NIPS*, pages 1141–1149, 2009.

[28] A. C. Lozano, G. Swirszcz, and N. Abe. Grouped orthogonal matching pursuit for variable selection and prediction. In *NIPS*, pages 1150–1158, 2009.

[29] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[30] V. G. Maz'ya and T. O. Shaposhnikova. Theory of multipliers in spaces of differentiable functions. *Russ. Math. Surv.*, 38(23), 1983.

[31] P. McCullagh and J. A. Nelder. *Generalized linear models (Second edition)*. London: Chapman & Hall, 1989.

[32] L. Meier, S. van de Geer, and P. Buhlmann. High-dimensional additive modeling. *Ann. Statist.*, 37(6B):3779–3821, 12 2009.

[33] S. K. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–32, 1994.

[34] P. Ravikumar, J. Lafferty, H. Liu, and L. Wasserman. Sparse additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(5):1009–1030, 2009.

[35] N. Segata and E. Blanzieri. Fast and scalable local kernel machines. *Journal of Machine Learning Research*, 11:1883–1926, 2010.

[36] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14:199–222, 2004.

[37] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

[38] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.

[39] A. K. Y. Truong. Fast growing and interpretable oblique trees via logistic regression models. *DPhil Thesis, University of Oxford*, 2009.

[40] B. Ustun, S. Tracà, and C. Rudin. Supersparse linear integer models for predictive scoring systems. In *AAAI Late Breaking Track*, 2013.

[41] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[42] J. Wang and V. Saligrama. Local supervised learning through space partitioning. In *NIPS*, pages 91–99, 2012.

[43] R. Wong. *Asymptotic approximations of integrals*. Computer science and scientific computing. Academic Press, Boston, San Diego, 1989. Includes indexes.

[44] Z. E. Xu, M. J. Kusner, K. Q. Weinberger, and M. Chen. Cost-sensitive tree of classifiers. In *ICML*, pages 133–141, 2013.

[45] J. Zhu and T. Hastie. Kernel logistic regression and the import vector machine. In *NIPS*, pages 1081–1088, 2001.

用可解释性换取准确性：斜树稀疏可加模型王家雷藤卷良平本桥洋介芝加哥大学 NEC 实验室美国 NEC 公司

jialei@uchicago.edu rfujimaki@nec-labs.com y-motohashi@bk.jp.nec.com

抽象的模型可解释性已被认为在实际数据挖掘中发挥着关键作用。可解释模型提供了对数据和模型行为的重要见解，并可能说服最终用户采用某些模型。然而，作为这些优势的回报，通常存在牺牲准确性，即需要限制模型表示的灵活性（例如线性、基于规则等）和模型复杂性，以便用户能够理解结果。本文提出倾斜树稀疏加性模型 (OT-SpAM)。我们的主要重点是开发一种模型，该模型牺牲一定程度的可解释性来提高准确性，但通过核支持向量机 (SVM)等完全非线性模型实现完全足够的准确性。OT -SpaAM 是区域特定预测模型的实例。它们将特征空间划分为具有稀疏倾斜树分裂的区域，并将局部稀疏可加专家分配给各个区域。为了保持 OT-SpAM 可解释性，我们必须保持整体模型结构简单，这产生了稀疏倾斜区域结构和稀疏局部专家的同时模型选择问题。我们通过扩展分解渐近贝叶斯推理来解决这个问题。我们在模拟、基准和现实世界中进行了演示就准确性而言，OT-SpAM 优于最先进的可解释模型，并且与内核 SVM 具有竞争力，同时仍然提供高度可理解的结果。

关键词可解释模型、模型选择、稀疏性一、简介模型可解释性已被认为在实际数据挖掘中发挥着关键作用。可解释模型提供了对数据和模型行为的重要见解，并可能说服最终用户采用某些模型。众所周知，尽管机器学习方法，例如内核机器[41,45]、提升[13]、随机森林[3]和深度神经网络[19,24]、简单模型，例如线性回归或允许免费制作本作品的

决策树在营销、医学分析和科学等应用中仍然受到青睐，对于这些应用来说，理解数据背后的现象比简单的准确预测更重要。然而，作为可解释性优势的回报，由于模型表示的灵活性（例如线性、基于规则等）和模型复杂性需要受到限制，以便用户能够理解结果，因此通常会牺牲准确性。

关于模型可解释性问题的讨论有两个关键概念：1）模型表示和 2）模型复杂性。对于前者，线性模型（例如广义线性模型（GLM）[31]）和决策树（例如，分类和回归树（CART）[4]）可能被认为是最容易解释的。尽管其模型表示的简单性有助于最终用户的理解，但它也限制了它们的预测能力。后者，特征稀疏性是提高线性模型可解释性的关键概念；即，选择少量关键特征使理解模型变得更加容易。此外，决策树的深层规则链可能会提高复杂数据的预测准确性，它使得规则结构难以理解。准确性和可解释性之间的权衡仍然是一个重要问题。本文提出了倾斜树稀疏加性模型（OT-SpAM），它提供了比线性模型和决策树更灵活的表示（因此，牺牲了一定程度的可解释性）。同时提供与此类完全模型相同的精度。非参数模型作为核支持向量机（KSVM），它们仍然保持易于解释的模型结构。OT-SpAM 是区域特定预测模型的实例，由区域指定器和区域特定预测器组成；指定者将特征空间划分为不相交的子空间（区域），并且各个预测器在相应的子空间中执行预测（正如我们在第 2 节中注意到的那样，区域特定的预测模型统一了上述两个可解释模型系列). OT-SpAM 采用倾斜树分割模型作为区域指定器，并采用稀疏加性模型作为各个区域特定的预测器。

如上所述，控制模型复杂度是维持模型可解释性的重要问题。对于 OT-SpAM，树结构（树的深度、区域数量等）、倾斜区域分裂的特征选择以及特征选择-局部稀疏专家的选择必须同时确定。我们通过利用分解渐近贝叶斯（FAB）推理来解决这个具有挑战性的模型选择问题[11,14]。通过类似 EM 的迭代优化，我们能够自动获得紧凑和可解释的 OT-SpaAM。我们在模拟、基准和现实世界数据集上证明，在准确性方面，OT-SpAM 优于最先进的可解释模型

DOI：http:/dx.doi.org/10.1145/2783258.2783407。

并与内核 SVM 相媲美，同时仍然提供易于理解的结果。

本文的其余部分组织如下。第 2 节提供了特定区域预测模型的文献综述。第 3 节和第 4 节分别介绍了 OT-SpAM 和提出的学习算法。模拟研究（第 5 节）和基准评估评估（第 6 节）定量地显示了 OT-SpAM 的优势，我们在第 7 节中展示了现实世界 POS（销售点）数据的结果。

2。文献综述本节主要关注区域特定的预测模型。表 1 总结了区域特定模型的特征，如下所述。对可解释模型的一般性和更广泛的调查可以在[12]中找到。

最简单的例子之一是线性模型，它只有一个全局区域，并采用线性预测模型作为区域特定的预测器。之前的一些研究[18,21]认为线性模型的倾斜超平面可能很难特征稀疏性是试图缓解此问题的一个关键概念，即选择少量关键特征可以使理解模型变得更加容易。为了获得稀疏线性模型（SLM），有多种方法，包括凸方法-ods（例如 Lasso [37]，L1 正则化逻辑回归[45]）和贪婪优化（例如正交匹配追踪[29,38]）已经被提出，尽管它们的主要焦点是模型泛化（减轻过度拟合），而不是增强模型的可解释性。稀疏加性模型（SAM）[20,32,34]引入特征非线性以提高精度。通过限制单个特征的非线性（即忽略非线性交互作用）特征之间），我们仍然可以可视化它们在特征方面（但非线性）的贡献，并从 SAM 中获得见解。SAM 的变体（核密度逻辑回归 (DLR)[6]和快速 fux 判别式 (FFD)[7]）已被在最近的 KDD 会议上提出了准确且可解释的模型，并且该方向的研究已成为社区密切关注的话题。

决策树，例如 CART，具有树结构的区域说明符，并使用各个区域中的常数值（也称为分段常数预测器）执行预测。倾斜决策树 [33]将区域说明符从单特征阈值扩展到线性超平面，并且贝叶斯树线性模型 (BTLM-s)[8]对区域特定的预测器使用线性超平面。通过空间分区的本地监督学习（LSL-SP）[42]对区域特定的预测器和区域指定器使用线性超平面。尽管这样的模型改进了预测 AC-简单决策树的准确性，其密集的线性超平面使模型难以理解。[44]研究了一种稀疏树模型，旨在减少测试时间成本。Eto 等人[11]提出了一种分层混合的变体采用因子分解渐近贝叶斯推理进行模型选择（FAB/HME）的专家模型。使用 FAB 框架 [14]，它们对特定区域的线性预测器实施稀疏性，这显着提高了密集线性预测器的可解释性，尽管它们的区域指定者的单特征阈值仍然限制了整体预测能力。超稀疏线性整数模型及其变体[26,40]也学习高度稀疏和可解释的模型结构，这也作为 KDD 2014 工业和政府轨道邀请演讲提出一系列局部线性模型（快速局部 KSVM [35]、局部线性 SVM [25]、集群 SVM [15]和局部深度核学习 (LDKL)[23]）使用测试点特定的线性预测器。它们没有明确的区域，而是在 fy 上生成线性预测器。对于我们的目的而言，这种方法的一个主要缺点是它们只能为每个测试点提供模型信息，这使得难以理解整体预测行为。

3.OT-SPAMS：斜树稀疏可加模型本节介绍 OT-SpAM 的详细信息。我们首先描述 OT-SpAM 的区域指定器和区域特定预测器，然后推导分解渐近贝叶斯推理，以解决同时模型选择的挑战。

3.1 OT-垃圾邮件我们的 OT-SpAM 是 HME 的变体 [22]，它是专家模型的树结构概率混合。在 HME 中，区域特定的预测器（树中的叶节点）被称为专家模型

n}N

s.假设我们有观测值{x n ,y n=1 ～X ×Y，其中 X∈RD 是协变量的域，Y∈R（对于回归任务）或{0,1}（对于分类任务），N 是样本数，D 为数据维数。树中的每个门（非叶子节点）决定一个数据实例是否会走向其左分支或右分支。在第 i 个门（i =1,.,G ，其中 G 是门的数量），令 zi ∈{0,1}为二元变量，指示实例 x 应该向下走哪个分支（不失一般性，令 zi =0 代表向左走的实例）。OT -SpaAM 采用以下逻辑超平面作为其倾斜区域说明符：

1.滋滋

11

p(zi|x)=ii ，

1 +exp(.w•x)1 +exp(w•x)

1）

其中 w i 预计会稀疏以保持可解释性。

令 zn =(z1 n ,.,zEn) ε {0,1}E（E 为专家数量）表示 x n 所属专家的指标，其中 zjn =1 代表属于 j 的实例-第 i 个专家。设 Gi 为第 i 个门的索引集，其中 Gi 包含第 i 个门的子树上的专家索引。设 Ej 为第 j 个专家的索引集，其中 Ej 包含从根到第 j 个专家的路径上的门的索引。给定区域说明符

i}G
超平面{w i=1,zn 上的分布可以描述如下：
乙
ni}Gn j
p(zn|x ,{w i=1)=ψ(x ,i,j)n ,(2)
j=1 i∈Ej
其中 ψ(x n,i,j)是 x n 在第 i 个门进入第 j 个专家所属分支的概率，更具体地说：
n j ∈Gleft
p(zi =0|x n),i
ψ(xn,i,j)=.(3)
p(zin =1|x n)，否则其中 Gi left 是第 i 个门的左子树中专家的索引集。
让我们考虑以下 SAM：
D
fj(x)=fjd(xd),(4)d=1
其中 fjd()是任何平滑单变量函数，其中许多函数预计为零（即稀疏）。请注意，如果我们用线性系数 θd 设置 fjd(xd)=θdxd，则 (4)将简化为标准线性模型。第 j 个专家的 y 生成分布由下式给出：
p(y|x,φj)=N (fj (x).y,σj 2),(5)
表 1：区域特定预测模型的比较（sp.=稀疏性，s.f.=单一特征，f.w.=特征明智）。SLM SAM DT ODT BTLM FAB/HME LSL-SP OT-SpAM LDKL
区域全局 s.f.threshold 倾斜 s.f.threshold 倾斜测试点特定区域 sp.X ×X ×X 不适用预测器线性 f.w.非线性常量线性 f.w.非线性线性预测因子 sp.X ×X ×X ×
参考文献[37][6,7,34][4][33,39][8][11][42]本文[23]
对于回归，其中 φj =(fj ,σj),和 1.y y
1 指数 (fj (x)
p(y|x,φj)=,1 +exp(fj (x)1+exp(fj (x)
6）
对于分类，其中 φj =fj 。总之，整个似然由下式给出：
n}Nn}Ni}G
p({y n=1|{x n=1,{φj }Ej=1,{w i=1)=(7)NE
妮妮
p(y |x ,φj)p(ζjn|x ,{w }iG =1).n=1 j=1
3.2 使用 FAB 框架的 OT-SpAM 模型选择为了学习 OT-SpAM 以及参数估计，我们必须同时解决三个模型选择问题：

M1：树形结构（门数、专家数等）。

M2：区域规范的稀疏性（逻辑门在（1）中提出。

M3：稀疏加性专家的稀疏性。

为了完成这些模型选择任务，我们对 OT-SpAM 使用 FAB 推理 [14]。请注意，FAB 最近已用于学习树状稀疏线性模型 [11]，本文将其框架扩展到 OT-SpAM 的学习 SpAM。

FAB 推理最大化以下贝叶斯边际对数似然：

恩

N}N

p({y n=1|{x n=1)=(8)

n}Nn}N

p({y n=1,{zn}Nn=1|{x n=1)

最大方程对数，

qq({zn}N)

n=1

其中 q 是 {zn}Nn=1 和最优 n}Nn}N 上的任意分布

q 是 q 。({zn}nN =1)=p({zn}nN =1|{y n=1,{x n=1)。设 =[W,]其中 W =[w 1 ,.,w G]和 =[Φ1,., ΦE]。然后将拉普拉斯方法 [43]应用于 (8)中对数函数内的分子，如下所示：

p({y}N ¯

n=1,{z}Nn=1|{x}Nn=1,{z}Nn=1,)

n=1)≈p({y}Nn=1|{x}N

D° j

我重力

2π)2 (2π)2

D

N w i D° j ,

氮

i=1 (zn)2 |F ¯ i |1/2 j=1 zn 2 | ¯ |1/2

n=1 j∈Gi j w (n=1 j)F° j

9）

在哪里

n

1 .2 log p(zn|x ,{w i}iG =1) ¯
F i =.iT ,(10)
wN zn
无线网络
n=1 j∈Gi j n
1 .2 log p(y n|zn ,x ,Φj)
¯
F° j =.N .(11)
δn .Φj .ΦTj
n=1j
¯ =[W¯ ,]¯ 是最大完全似然估计量，D.表示的维数。
尽管 Eto 等人[11]利用大数定律渐近地忽略了|F¯ w i |1/2 和|F¯ ° j，但本文考虑以下上限以获得更好的近似，利用 Hadamard 不等式[30]：
2.
D
¯i |1/2F w ≤N .zn .j n=1 j∈Gi w i N .n=1 j∈Gi zn nj .2 log Ψ(x ,i,j).2(wi •xn)
12）
2 2
¯|1/2F° j ≤N D° j zn j n=1 N .2 nD° jlog p(y n|x ,Φj).2fjn=1 。
13）
将式（9)、(12)、(13）代入式（8)，得到分解信息准则（FIC）如下：
FIC({x,y}Nn=1,)=max n=1,q),(14)
L({x,y}N
q,
在哪里
L({x,y}Nn=1,q)=Eqn=1,{z}Nn=1,)
log p({y}Nn=1|{x}N
GN EN
w i|0 |fj |0
log(αnij zjn).log(ηjn zjn)
22