Fully-Automatic Bayesian Piecewise Sparse Linear Models

Riki Eto* **Ryohei Fujimaki†** **Satoshi Morinaga*** **Hiroshi Tamano***
*NEC Corporation †NEC Laboratories America
{r-eto@bc, morinaga@cw, h-tamano@bx}.jp.nec.com rfujimaki@nec-labs.com

Abstract

Piecewise linear models (PLMs) have been widely used in many enterprise machine learning problems, which assign linear experts to individual partitions on feature spaces and express whole models as patches of local experts. This paper addresses simultaneous model selection issues of PLMs; partition structure determination and feature selection of individual experts. Our contributions are mainly three-fold. First, we extend factorized asymptotic Bayesian (FAB) inference for hierarchical mixtures of experts (probabilistic PLMs). FAB inference offers penalty terms w.r.t. partition and expert complexities, and enable us to resolve the model selection issue. Second, we propose posterior optimization which significantly improves predictive accuracy. Roughly speaking, our new posterior optimization mitigates accuracy degradation due to a gap between marginal log-likelihood maximization and predictive accuracy. Third, we present an application of energy demand forecasting as well as benchmark comparisons. The experiments show our capability of acquiring compact and highly-accurate models.

1 Introduction

Piecewise linear models (PLMs) have been widely used in many enterprise machine learning problems [13, 14, 15] which assign linear experts to individual partitions on feature spaces and express whole models as patches of local experts. Depending on how they characterize partitions and their local linear experts, various PLMs have been studied such as decision/regression trees [1], hierarchical mixtures of experts (HMEs) [2], Bayesian treed linear models (BTLMs) [3], and more advanced region-specific linear models [9, 10]. For example, regression trees employ partitions specified by rule chains and constant-valued regressors while the model proposed by [9] employs linear partition specifiers and linear classifiers.

A practical requirement to PLMs is compact and interpretable representation in terms of both partitions and experts as well as high predictive performance. If the partition structure is too complex (e.g., decision tree with deep rule chains), it's very difficult to understand the meaning of the model, and it eventually loses important advantage (interpretability) over highly nonlinear models like kernel machines [18, 19]. Furthermore, it is significant to make individual linear experts sufficiently sparse since locally-significant features may be different among local regions and capturing such a local feature structure gives us better understanding about data generation processes. We refer to such PLMs with sparse linear experts as piecewise sparse linear models (PSLMs). For learning PSLMs, a challenging model selection problem must be addressed; simultaneous partition-structure determination and feature selection for individual experts. To the best of our knowledge, this simultaneous optimization has not yet been well studied and remained to be accomplished.

This paper addresses the above described model selection issue of PSLMs, and our contributions are mainly three-fold as summarized below.

FAB Inference for HMEs This paper employs HMEs, which is one of the most well-studied probabilistic piecewise models [16, 17], with probabilistic rule-based partitions. We extend factorized asymptotic Bayesian (FAB) inference for HMEs, which is a recently-developed Bayesian approximation inference for latent variable models such as mixture models [4], hidden Markov models [5], and latent feature models [6]. In FAB/HMEs, two L_0-regularization effects are naturally induced: 1) structure-level regularization

Appearing in Proceedings of the 17^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33.

which penalizes the complexity of the partition structure, and 2) expert-level regularization which enforces sparseness to individual experts. By the combination of the shrinkage step in FAB procedure and an advanced L_0 greedy optimization algorithm (the forward-backward greedy algorithm; FoBa [7]), FAB/HMEs simultaneously identify the partition structure and sparseness of individual experts. It is worth noticing that the optimization of FAB/HMEs starts from the sufficient number of experts (e.g., $\mathcal{O}(\log N)$, where N is the number of samples.) and there is no sensitive tunable hyper-parameters, and we can obtain PSLMs in a fully-automatic manner.

Improved Posterior Optimization We develop an algorithm to mitigate a gap between the FAB objective function (factorized information criterion; FIC) and our true objective one (predictive error). Roughly speaking, the gap arises because the model optimization is done on posterior expectation while prediction is done using prior. We introduce a projection step in posterior optimization, which minimizes distance between posterior and prior, by keeping guarantee of FIC monotonic increase over FAB EM iterations. In addition, we completely match posterior and prior as post-processing. We show that these two heuristics significantly improve predictive accuracy in comparison with naive FAB/HMEs.

Application to Energy Demand Forecasting We demonstrate capability of FAB/HMEs in application to a building energy demand forecasting problem. We show that FAB/HMEs offer a simple but highly-accurate forecasting model, which captures reasonable energy usage trend. In addition to energy demand forecasting, we present experimental results on benchmark and simulation datasets which show the superiority of FAB/HMEs over the other PLMs.

2 Related Work

Piecewise Linear Models The most well-studied precursors of PLMs are regression tree models (RegTrees) [1], in which partitions are specified by chains of rules and local experts are constant-valued. Although the representation of RegTree is highly interpretable, predictive abilities of individual experts are not high and the tree depth tends to be large in order to achieve good prediction performance. HMEs [2] adopt a divide-and-conquer strategy for constructing piecewise models. Their treed-partition structure is determined by "gating" functions, which are probabilistic soft-partitioning functions. Although HMEs can express any partition structures by designing the (probabilistic) gating functions, this paper employs a rule-chained (treed) partition structure so that the learned partition structure is practically understandable. BTLMs [3] are also one of the divide-and-conquer models and have hard-partition structures like RegTree, but local experts represent generalized linear models. The partition structures of BTLMs are explored by the Markov chain Monte Carlo search.

Recently, the algorithms named local supervised learning through space partitioning (LSL-SP) [9] and cost-sensitive tree of classifiers (CSTC) [10] are developed. LSL-SP is only available for classification. The target of CSTC is test-time speed up rather than our target (PSLMs as interpretable and accurate models.)

FAB for Mixture Models Suppose we have observation data $x^N = x^{(1)}, \cdots, x^{(N)}$ where $x^{(n)} \in \mathbb{R}^D$. Let us denote latent variable corresponding to x^N as $z^N = z^{(1)}, \cdots, z^{(N)}$ where $z^{(n)} \in \{1, 0\}^C$ is an indicator vector which means that a mixture component generates $x^{(n)}$. FIC is derived as an asymptotic approximation of marginal log-likelihood as follows:

$$FIC^{mm} := \max_q \Big\{ \mathbb{E}_q \Big[\log p(x^N, z^N|\bar{\theta}) - \frac{D_Z}{2} \log N - \sum_k \frac{D_k}{2} \log \sum_n z_k^{(n)} \Big] + H(q) \Big\}, \quad (1)$$

where $q(z^N)$ is a variational distribution on z^N; and θ is a model parameter; $\bar{\theta}$ is the maximum likelihood estimator (MLE) of $p(x^N, z^N|\theta)$[1]; D_Z and D_k are the parameter dimensionalities of $p(z^N)$ and $p_k(x^N|z_k^N)$; $H(q)$ is the entropy of $q(z^N)$. The most important term in FIC^{mm} (1) is $\log(\sum_n z_k^{(n)})$, which offers such theoretically desirable properties for FAB inference as automatic shrinkage of irrelevant latent variables and parameter identifiability [4].

Direct optimization of FIC^{mm} is difficult because: **(i)** evaluation of $\mathbb{E}_q[\log \sum_n z_k^{(n)}]$ is computationally infeasible, and **(ii)** the MLE is not available in practice. Instead, FAB optimizes a tractable lower bound of an FIC [4]. For **(i)**, since $-\log \sum_n z_k^{(n)}$ is a convex function, its linear approximation at $Nr_k > 0$ yields the lower bound $-\log \sum_n z_k^{(n)} \geq -\{\log Nr_k + (\sum_n z_k^{(n)}/N - r_k)/r_k\}$, where $0 < r_k \leq 1$ is a linearization parameter. For **(ii)**, since, from the definition of the MLE, the inequality $\log p(x^N, z^N|\bar{\theta}) \geq \log p(x^N, z^N|\theta)$ holds for any θ, we optimize θ along with q. Alternating maximization of the lower bound with respect to q, θ, and r_k guarantees a monotonic increase in the FIC lower bound [4].

[1] While $p(x^N|\theta)$ is a non-regular model, $p(x^N, z^N|\theta)$ is a regular model[22].

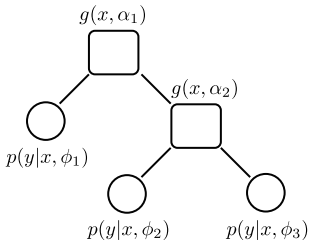

Figure 1: An example of HMEs model.

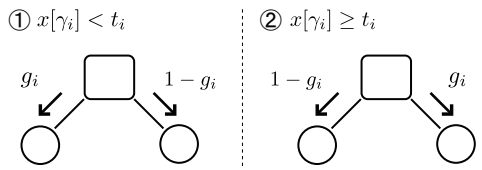

Figure 2: An example of Bernoulli Gating Node.

3 Piecewise Sparse Linear Regressors with FAB/HMEs

3.1 Hierarchical Mixtures of Experts

HMEs models are extensions of mixture of experts models [20] and are represented in tree structures, in which experts are mixed according to gating functions. Fig. 1 shows a rough sketch of a HMEs model. Inputting x to the top gating node, the gating nodes (squared nodes) select an appropriate expert node (a circled node) for prediction, in a soft decision tree manner.

The i-th gating node has an associated binary variable $z_i \in \{0, 1\}$ of x and y, where $z_i = 1$ means the data is generated from one of experts in the left-side branches, and $z_i = 0$ vice versa. As far as we know, most studies using HMEs have employed the sigmoid gating function. However, for learning HMEs, we typically need EM-like iterative optimization, and therefore the sigmoid function is relatively computationally-expensive to optimize in each single iteration. Alternatively, this paper employs the following Bernoulli gating function:

$$g(x, \alpha_i) := g_i U(t_i - x[\gamma_i]) + (1 - g_i) U(x[\gamma_i] - t_i), \tag{2}$$

where $g_i \in \mathbb{R}$ satisfy $[0, 1]$, U is the step function, γ_i is the index w.r.t. the elements of x, $t_i \in \mathbb{R}$ is an arbitrary value and $\beta_i = \{g_i, t_i\}$. For example, when $x[\gamma_i] < t_i$, $g(x, \alpha_i) = g_i$ and otherwise $g(x, \alpha_i) = 1 - g_i$ as shown in Fig. 2. The probability of z_i is given by:

$$p_g(z_i|x, \beta_i; \gamma_i) = g(x, \alpha_i)^{z_i} [1 - g(x, \alpha_i)]^{1 - z_i}, \tag{3}$$

where the function $g(x, \alpha_i)$ is referred to as the gating function parameterized by $\alpha_i = (\beta_i, \gamma_i)$. Starting at the top of the tree, we thereby stochastically choose a path down to the j-th expert node, and then generate a target value by the probabilistic model explained next.

The conditional distribution of the j-th expert (linear regression regressors, i.e., $y = w_j^T x + \varepsilon$ with i.i.d. Gaussian noise $\varepsilon \sim \mathcal{N}(\varepsilon|0, \sigma_j^2)$) is given by $p(y|x, \phi_j)$, that is,

$$p(y|x, \phi_j) = \mathcal{N}(y|w_j^T x, \sigma_j^2), \tag{4}$$

where $\phi_j = (w_j, \sigma_j^2)$. We here omit the bias term for notational simplicity. Let us denote regression target as $y^N = y^{(1)}, \cdots, y^{(N)}$ where $y^{(n)}$ corresponds to $x^{(n)}$.

Let us here define a few notations. Let us first introduce the i-th gating index set, $\mathcal{G}_i$ $(i = 1, \ldots, G)$, and the j-th expert index set, $\mathcal{E}_j$ $(j = 1, \ldots, E)$. $\mathcal{G}_i$ contains all indices of the expert nodes on the sub-tree of the i-th gating node. $\mathcal{E}_j$ contains all indices of the gating nodes on the unique path from the root node to the j-th expert node, which is called the j-th path. For example, in Fig. 1, $\mathcal{G}_2 = \{2, 3\}$ and $\mathcal{E}_3 = \{1, 2\}$. Also, let us define a function ψ as follows:

$$\psi(a, i, j) := \begin{cases} a & \text{if the } j \text{ is in the left sub-tree of } i \\ 1 - a & \text{otherwise} \end{cases},$$

$$\psi_g(x, i, j) := \psi(g(x, \alpha_i), i, j).$$

Then, HMEs models are defined as follows:

$$p(y|x, \theta; \gamma) = \sum_{j=1}^{E} \prod_{i \in \mathcal{E}_j} \psi_g(x, i, j) p(y|x, \phi_j), \tag{5}$$

where $\theta = (\beta_1, \ldots, \beta_G, \phi_1, \ldots, \phi_E)$ and $\gamma = (\gamma_1, \ldots, \gamma_G)$ represents models of gating nodes. We define the latent variable related to the j-th path as:

$$\zeta_j := \prod_{i \in \mathcal{E}_j} \psi_z(i, j) \in \{0, 1\}, \tag{6}$$

where $\psi_z(i, j) := \psi(z_i, i, j)$. We get conditional distributions as follows:

$$p(y^N|\zeta^N, x^N, \phi) = \prod_{j=1}^{E} p(y^N|x^N, \phi_j)^{\zeta_j^N}, \tag{7}$$

$$p(\zeta^N|x^N, \beta; \gamma) = \prod_{n=1}^{N} \prod_{j=1}^{E} \prod_{i \in \mathcal{E}_j} \psi_g(x^{(n)}, i, j)^{\zeta_j^{(n)}}, \tag{8}$$

where $\phi = (\phi_1, \ldots, \phi_E)$, $\beta = (\beta_1, \ldots, \beta_G)$. Prediction is done through the gating functions as follows:

$$\hat{y} = w_{j^*}^T x \quad \text{where} \quad j^* = \arg\max_j \prod_{i \in \mathcal{E}_j} \psi_g(x, i, j). \tag{9}$$

3.2 FIC and FAB for HMEs

By following a standard derivation procedure of FIC, we obtain FIC^{hme} as follows:

$$FIC^{hme}(\theta, q) = \max_{q} \Big\{ \mathbb{E}_q \Big[\log p(y^N, \zeta^N | x^N, \bar{\theta}) - \sum_{i=1}^{G} \Big(\frac{D_{\beta_i}}{2} \log \sum_{n=1}^{N} \sum_{j \in \mathcal{G}_i} \zeta_j^{(n)} \Big) - \sum_{j=1}^{E} \frac{D_{\phi_j}}{2} \log \sum_{n=1}^{N} \zeta_j^{(n)} \Big] + H(q) \Big\}. \tag{10}$$

As with the case of mixture models (1), FIC^{hme} is not available in practice, and we employ the lower bounding techniques (i) and (ii). FAB E-step and M-step are derived as follow by maximizing the lower bound of FIC^{hme}. Hereinafter, we refer to the lower bound as FIC^{hme}_{LB} or merely FIC_{LB} and let the superscription (t) represents the t-th iteration.

Note that the following FAB EM iteration monotonically increases FIC_{LB}, and therefore we employ $FIC^{(t)}_{LB} - FIC^{(t-1)}_{LB} < \delta$ as the condition of algorithm termination.

FAB E-step The E-step optimizes the variational distribution q as follows:

$$q^{(t)}(\zeta_j^{(n)}) \propto \prod_{i \in \mathcal{E}_j} \psi_g^{(t-1)}(x^{(n)}, i, j) p(y^{(n)} | x^{(n)}, \phi_j^{(t-1)}) \exp \Big\{ \sum_{i \in \mathcal{E}_j} \frac{-D_{\beta_i}}{2N_{\beta_i}^{(t-1)}} + \frac{-D_{\phi_j}}{2N_{\phi_j}^{(t-1)}} \Big\}, \tag{11}$$

where $\psi_g^{(t)}(x^{(n)}, i, j) = \psi(g(x^{(n)}, \alpha_i^{(t)}), i, j)$, $N_{\beta_i}^{(t)} = \sum_{n=1}^{N} \sum_{j \in \mathcal{G}_i} q^{(t)}(\zeta_j^{(n)})$ and $N_{\phi_j}^{(t)} = \sum_{n=1}^{N} q^{(t)}(\zeta_j^{(n)})$. Similar to the other FAB algorithms [4, 5, 6], the FAB unique regularization effect appears as exponentiated form, i.e., $\exp \Big\{ \sum_{i \in \mathcal{E}_j} \frac{-D_{\beta_i}}{2N_{\beta_i}^{(t-1)}} + \frac{-D_{\phi_j}}{2N_{\phi_j}^{(t-1)}} \Big\}$. This regularization penalizes smaller and more complex experts, and eventually such experts are likely to become smaller. This makes it possible to prune the tree-structure naturally.

FAB M-step The M-step optimizes the models of expert and gating nodes: S, γ and the parameter θ as follows:

$$\gamma_i^{(t)}, \beta_i^{(t)} = \arg\max_{\gamma_i, \beta_i} \Big\{ \sum_{n=1}^{N} \sum_{j \in \mathcal{G}_i} q^{(t)}(\zeta_j{}^{(n)}) \log \psi_g(x^{(n)}, i, j) - \frac{D_{\beta_i}}{2} \log(N_{\beta_i}^{(t)}) \Big\}, \tag{12}$$

$$S_j^{(t)}, \phi_j^{(t)} = \arg\max_{S_j, \phi_j} \Big\{ \sum_{n=1}^{N} q^{(t)}(\zeta_j^{(n)}) \log p(y^{(n)} | x^{(n)}, \phi_j) - \frac{\mathcal{D}_{\phi_j}}{2} \log(N_{\phi_j}^{(t)}) \Big\}. \tag{13}$$

Algorithm 1 Optimization of Bernoulli Gating function

Input: x^N, $q^{(t)}(\zeta_j^{(n)})$
Output: $\gamma_i^{(t)}, \beta_i^{(t)}$
1: **for** γ_i in the dimension of x^N **do**
2: **for** t_i in the domain of x^N **do**
3: Calculate $g_i^{(t)}$ by (16).
4: **end for**
5: **end for**
6: Choose $\gamma_i^{(t)}, \beta_i^{(t)}$ by (15).

About details of these optimization, we describe in the subsections 3.3 and 3.4.

FAB Shrinkage Step As is shown in FAB E-step, smaller and more complex paths are likely to become smaller. FAB removes such "non-effective" experts from the model as follows:

$$q^{(t)}(\zeta_j^{(n)}) = \begin{cases} 0 & \text{if } N_{\phi_j}^{(t)} < \delta \\ q^{(t)}(\zeta_j^{(n)}) / Q_j^{(t)} & \text{otherwise} \end{cases}, \tag{14}$$

where δ and $Q_j^{(t)}$ are a threshold value and a normalization constant for $\sum_{j=1}^{E} q^{(t)}(\zeta_j^{(n)}) = 1$.

This shrinkage step addresses one of our model selection issues, i.e., partition structure determination. Starting from a symmetric tree with a sufficiently large number of experts, FAB/HMEs gradually remove irrelevant experts in this shrinkage step.

3.3 Gating Function Optimization

Algorithm 1 solves (12) which is rewritten as follows:

$$\gamma_i^{(t)}, \beta_i^{(t)} = \arg\max_{\gamma_i, \beta_i} \Big[\sum_{n \in \mathcal{T}_{li}} \Big\{ \sum_{j \in \mathcal{G}_{iL}} q^{(t)}(\zeta_j^{(n)}) \log(1 - g_i) + \sum_{j \in \mathcal{G}_{iR}} q^{(t)}(\zeta_j^{(n)}) \log g_i \Big\} + \sum_{n \in \mathcal{T}_{si}} \Big\{ \sum_{j \in \mathcal{G}_{iL}} q^{(t)}(\zeta_j^{(n)}) \log g_i + \sum_{j \in \mathcal{G}_{iR}} q^{(t)}(\zeta_j^{(n)}) \log(1 - g_i) \Big\} \Big], \tag{15}$$

where $\mathcal{T}_{li}, \mathcal{T}_{si}$ are the sets of samples whose the γ_i-th dimension is larger or smaller than t_i, $\mathcal{G}_{iL}$ contains all indices of the expert nodes on the left sub-tree of the i-th gating node and $\mathcal{G}_{iR}$ is similarly defined for the right sub-tree of the i-th gating node. At line3, (15) has the analytic solutions with given γ_i and t_i as follows:

$$g_i^{(t)} = \frac{1}{N_{\beta_i}^{(t)}} \Big\{ \sum_{n \in \mathcal{T}_{si}} \sum_{j \in \mathcal{G}_{iL}} q^{(t)}(\zeta_j^{(n)}) + \sum_{n \in \mathcal{T}_{li}} \sum_{j \in \mathcal{G}_{iR}} q^{(t)}(\zeta_j^{(n)}) \Big\}. \tag{16}$$

In practical implementation, we calculate the attribute-wise sort before the FAB EM itera-

tion ($\mathcal{O}(N \log N)$). Then, (16) can be solved with $\mathcal{O}(N)$ time complexity.

3.4 Sparse Optimization of Linear Regressors

In (13), D_{ϕ_j} is the dimensionality of ϕ_j given S_j and can be rewrite using the L_0-norm of ϕ_j as $D_{\phi_j} = ||w_j||_0 + 1$ (we should add the dimensionality of σ_j^2). Then, we transform (13) to the following feature selection problem with the sparse constraint:

$$w_j^{(t)} = \arg\min_{w_j} Q(w_j, (\sigma_j^2)^{(t-1)}) \quad \text{s.t. } ||w_j||_0 \leq K, \tag{17}$$

$$\sigma_j^{2(t)} = \arg\min_{\sigma_j^2} Q(w_j^{(t)}, \sigma_j^2), \tag{18}$$

$$Q(w_j, \sigma_j^2) = -\sum_{n=1}^{N} q^{(t)}(\zeta_j^{(n)}) \log p(y^{(n)}|x^{(n)}, w_j, \sigma_j^2), \tag{19}$$

where K is a constant corresponding to the regularization term $\mathcal{D}_{\phi_j} \log(N_{\phi_j}^{(t)})/2$.

We solve the L_0-regularized feature selection problem (17) by applying the forward backward greedy (FoBa) algorithm [7, 8] since it offers the tightest upper bounds of feature selection error, estimation error, and objective error. Although the upper bounds for the original FoBa algorithm have been derived for (non-weighted) least square regression, we can achieve the same bounds by slightly modifying the proofs of [8] (we omit the details because of space limitation).

The FoBa algorithm of (17) is described in Algorithm 2 (for notational simplicity, we omit the index j in Algorithm 2). In Algorithm 2, the superscription (k) represents the iteration number of FoBa, $F^{(k)}$ is the index set of non-zero elements of $w^{(k)}$, e_i is the natural base of the i-th feature, and $\hat{\phi}(F^{(k)})$ is the minimizer of $Q(w_j, \sigma_j^2)$ under the constrains that the elements of w_j outside $F^{(k)}$ are zero. $\nabla_{w_j} Q(w_j, \sigma_j^2)$ is gradient of $Q(w_j, \sigma_j^2)$ w.r.t. w_j, i.e.,

$$\nabla_{w_j} Q(w_j, \sigma_j^2) = \sum_{n=1}^{N} q^{(t)}(\zeta_j^{(n)}) \frac{1}{\sigma_j^2}(w_j^T x^{(n)} - y^{(n)}) x^{(n)}. \tag{20}$$

Note that we do not need to manually set K in (17). In each M-step (and for each expert), FAB/HMEs automatically adjust the value of K, equivalently $\log(N_{\phi_j}^{(t)})/2$, depending on $q^{(t)}$.

3.5 Improving Posterior Optimization

Although the inference of the FAB/HME algorithm works as expected, it might have an issue for prediction. The predictive function (9) is defined to used a single expert which maximizes the "gating probability". However, in the FAB M-step, each expert is optimized using variational posterior (i.e., $q(\zeta^N)$). This gap between predictive objective and FIC maximization causes predictive performance degradation. This paper introduces two techniques for filling this gap. A key idea behind both techniques is to match the variational posterior $q(\zeta^N)$ and the gating probability $p(\zeta^N|x^N, \beta; \gamma)$.

Algorithm 2 FoBa for FAB M-step

Input: y^N, x^N, $q^{(t)}(\zeta^{(n)})$, σ^2
Output: w^*
Let $F^{(0)} = \emptyset$, $w^{(0)} = 0$, $k = 0$
while TRUE **do**
  %% forward step
  $i^{(k)} = \arg\max_{i \notin F^{(k)}} : |\nabla_w Q(w^{(k)}, \sigma^2)_i|$
  $F^{(k+1)} = F^{(k)} \cup \{i^{(k)}\}, w^{(k+1)} = \hat{\phi}(F^{(k+1)})$
  $k = k + 1$
  %% stopping determination
  if $Q(w^{(k-1)}, \sigma^2) - Q(w^{(k)}, \sigma^2) \leq \log(N_{\phi_j}^{(t)})/2$ **then**
    $k = k - 1$ and break
  end if
  %% backward step
  while TRUE **do**
    if $\min_i Q(w^{(k)} - w_i^{(k)} e_i, \sigma^2) - Q(w^{(k)}, \sigma^2) > \log(N_{\phi_j}^{(t)})/2$ **then**
      break
    else
      $i^{(k)} = \arg\min_i Q(w^{(k)} - w_i^{(k)} e_i, \sigma^2)$
      $k = k - 1$
      $F^{(k)} = F^{(k+1)} - \{i^{(k+1)}\}, w^{(k)} = \hat{\phi}(F^{(k)})$
    end if
  end while
end while
$w^* = w^{(k)}$

Projection E-step In stead of maximizing FIC_{LB} like (11), by following the idea of the generalized EM (GEM) algorithm [21], our new E-step updates the variational posterior such that $FIC_{LB}(\theta^{(t-1)}, q^{(t)}) > FIC_{LB}(\theta^{(t-1)}, q^{(t-1)})$ holds. Although the GEM algorithm is motivated to make optimization problems tractable, our new E-step is motivated to fix the gap between $q(\zeta^N)$ and $p(\zeta^N|x^N, \beta; \gamma)$.

Let us define $\rho = (\rho^{(1)}, \ldots, \rho^{(n)})$ and $q_\rho^{(t)}(\zeta^{(n)}) = \rho^{(n)} q^{(t)}(\zeta^{(n)})$. The following lemma describes a function which satisfies $FIC_{LB}(\theta^{(t-1)}, q^{(t)}) > FIC_{LB}(\theta^{(t-1)}, q^{(t-1)})$.

Lemma 1 *$FIC_{LB}(\theta^{(t-1)}, q_\rho^{(t)} + q_{1-\rho}^{(t-1)}) \geq FIC_{LB}(\theta^{(t-1)}, q^{(t-1)})$ holds with any $\rho^{(n)} \in [0, 1]$.*

Algorithm 3 Projection E-step

Input: x^N, $q^{(t-1)}$, $\theta^{(t-1)}$
Output: $q^{(t)}$
1: Calculate $q^{(t)}(\zeta^N)$ by (11).
2: Calculate $q^{(t)}_{gate}(\zeta^N) = p(\zeta^N|x^N, \beta^{(t)}; \gamma^{(t)})$ by (12).
3: Calculate $\rho^{(n)}$ by (21).
4: **if** $0 < \rho^{(n)} < 1$ **then**
5: Update $q^{(t)}(\zeta^{(n)})$ by (22).
6: **end if**

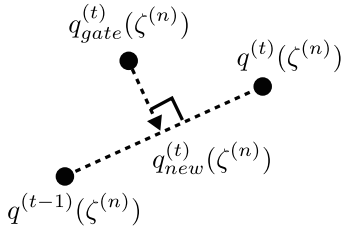

Figure 3: Geometric interpretation of the projection E-step.

Proof Because of additive form of FIC_{LB} in terms of q, we can decompose $FIC_{LB}(\theta^{(t-1)}, q^{(t)}_{\rho} + q^{(t-1)}_{1-\rho})$ as:

$$\begin{aligned} &FIC_{LB}(\theta^{(t-1)}, q^{(t)}_{\rho} + q^{(t-1)}_{1-\rho}) \\ &\quad = FIC_{LB}(\theta^{(t-1)}, q^{(t)}_{\rho}) + FIC_{LB}(\theta^{(t-1)}, q^{(t-1)}_{1-\rho}) \\ &\quad \geq FIC_{LB}(\theta^{(t-1)}, q^{(t-1)}_{\rho}) + FIC_{LB}(\theta^{(t-1)}, q^{(t-1)}_{1-\rho}) \\ &\quad = FIC_{LB}(\theta^{(t-1)}, q^{(t-1)}) \quad \blacksquare \end{aligned}$$

Algorithm 3 summarizes our new E-step which we refer to as the projection E-step. We first calculate $q^{(t)}(\zeta^N)$ and $q^{(t)}_{gate}(\zeta^N)$ (line1 and line2). Then, we calculate ρ by minimizing L_2 distance between $q^{(t)}(\zeta^N)$ and $q^{(t)}_{gate}(\zeta^N)$ as follows:

$$\rho^{(n)} = \frac{(q^{(t)}_{gate}(\zeta^{(n)}) - q^{(t-1)}(\zeta^{(n)}))^T (q^{(t)}(\zeta^{(n)}) - q^{(t-1)}(\zeta^{(n)}))}{\|q^{(t)}(\zeta^{(n)}) - q^{(t-1)}(\zeta^{(n)})\|_2^2}. \quad (21)$$

On the basis of Lemma 1, for all n satisfying $0 < \rho^{(n)} < 1$, we update $q^{(t)}(\zeta^{(n)})$ as follows:

$$q^{(t)}_{new}(\zeta^{(n)}) = \rho^{(n)} q^{(t)}(\zeta^{(n)}) + (1 - \rho^{(n)}) q^{(t-1)}(\zeta^{(n)}). \quad (22)$$

As Fig. 3 explains, $q^{(t)}_{new}(\zeta^{(n)})$ calculated by (22) is an orthogonal projection of $q^{(t)}_{gate}(\zeta^{(n)})$ onto the interval between $q^{(t)}(\zeta^{(n)})$ and $q^{(t-1)}(\zeta^{(n)})$.

Hard-Gate Post-Processing Although the projection E-step mitigates the gap, it cannot completely fix the gap. We employ a heuristic post-processing after the FAB EM iteration converges. The procedure is summarized as follows. First, we update the branching probabilities as follow:

$$g_i^{hard} = \begin{cases} 1 & \text{if } g_i > 0.5 \\ 0 & \text{otherwise} \end{cases}. \quad (23)$$

With g_i^{hard}, $q^{hard}(\zeta_j^{(n)}) = p(\zeta_j^{(n)}|x^{(n)}, \beta^{hard}; \gamma) \in \{1, 0\}$ is satisfied, where $\beta^{hard} = (g_i^{hard}, t_i)$. Then, with $q^{new}(\zeta_j^{(n)})$, we re-optimize the gates and experts by (12) and (13), and re-update the branching probabilities by (23). After this post-processing, the branching probabilities which values are 0 or 1 provide hard-partitioning, and therefore we refer to this procedure as hard-gate post-processing.

4 Experiments

We employed $\delta = 10^{-5}$ (termination condition) and $\epsilon = N \times 10^{-2}$ (shrinkage threshold). The number of initial experts was 32 (5-depth symmetric tree). Observation and target values were standardized in advance. We denote FAB/HMEs with the projection E-step and hard-gate post-processing as FAB/HMEs$^+$.

4.1 Artificial Simulation

We first demonstrate how the model selection of FAB/HMEs$^+$ works using simple artificial data. The true tree structure is described in Fig. 4, which has 5 experts and each expert uses 2-4 features. On the i-th gating node, g_i was fixed to 1, γ_i and t_i were randomly selected from $[1, D]$ and $[0, 1]$, respectively. On the j-th expert, the non-zero elements of w_j were randomly sampled from $[0, 1]$. x^N and y^N are sampled from Uniform$[0, 1]$ and $\mathcal{N}(y^{(n)}|w_j^T x^{(n)}, 0.1)$, where $N = 3000$ and $D = 10$.

Fig. 4 illustrates $FIC^{(t)}_{LB}$ (top) and estimated expert coefficient matrix (bottom) over the FAB EM iteration. At $t = 1$, 32 experts were randomly initialized. Over the EM iteration ($t = 5, 11$), FAB/HME$^+$ simultaneously selected a partition structure and expert sparseness, and it successfully recovered experts' signals and the tree partition structure at the convergence point ($t = 18$). $FIC^{(t)}_{LB}$ monotonically increases except two points (dashed circle and square) where the shrinkage operation and post-processing were conducted.

4.2 Benchmark Evaluation

We compared FAB/HMEs, on 9 regression datasets in LIACC repository [12] and UCI repository [11], with EM/HMEs, BTLMs[2], RegTree[3]. Also, support vector

[2]We used R package (tgp).
[3]We used Python package (scikit-learn).

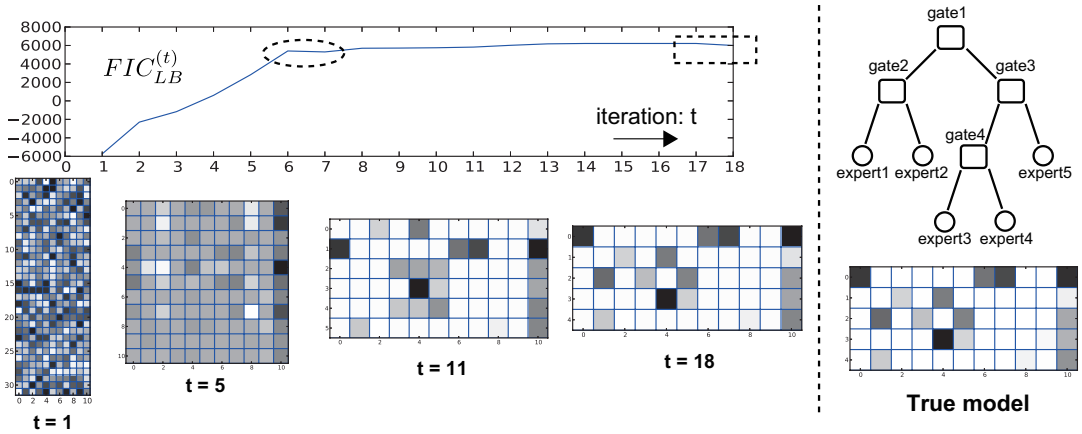

Figure 4: Model selection procedure of FAB/HMEs$^+$. Binary maps represent the (intermediate) estimated coefficients of experts, where the rows and columns mean the number of experts and that of features, respectively.

regression with RBF kernel (RBF-SVR)[3] [23] was compared in order to evaluate performance degradation from highly-non-linear regression model (in general, non-linear regression is better in performance than PLMs.) We used 2-loop cross validation with 10-fold outer loop for evaluating test RMSE and 3-fold inner loop for parameter selection. Note that FAB/HMEs do not need 2-loop cross validation and have advantage in computational efficiency while we did not evaluate it because of implementation difference.

Table 1 and 2 summarize test RMSEs and complexities of learned models. We observe:

- For all datasets, FAB/HMEs$^+$ was significantly better than FAB/HMEs and we confirmed the effect of our improved posterior optimization.
- FAB/HMEs$^+$ and BTLMs performed competitively with each other w.r.t. test RMSE. They performed better than EM/HMEs and RegTree.
- FAB/HMEs$^+$ obtained the most compact representations. The number of experts was much smaller than RegTree and expert cardinality was much smaller than BTLMs.
- Except kin8nm and pol, FAB/HMEs$^+$ performed slightly worse than RBF-SVR, but the difference was not significant. For puma32H, FAB/HME$^+$ performed even better than RBF-SVR.

In summary, the results indicate that FAB/HMEs$^+$ achieved close-to-best performance with the simplest model representation.

4.3 Energy Demand Forecasting

Motivation One of important and typical industrial applications of FAB/HMEs is demand forecasting. In real world demand forecasting projects, we are often required to explain how our demand forecasters predict target values as well as to achieve high prediction accuracy. For example, in energy demand forecasting which is demonstrated in this paper, we can control energy generation on the basis of forecasting results (high predictive accuracy). Further, the forecasting model itself would be useful for energy generation planning (interpretable model representation). FAB/HMEs' compact representation for the latter contribution is one of significant features in this kind of applications.

Table 2: Comparison of model complexities (average over 9 benchmark datasets).

	FAB/HMEs$^+$	BTLMs	RegTree
num of experts	9.7	12.7	97.0
cardinality	5.1	33.0	1.0
total	49.5	419.1	97.0

Data Description We collected energy consumption of an office building for which we would like to make hourly demand forecasting in order to control/plan energy generation. We have 14 attributes; day of the week (7 binary attributes), time (1 integer attribute ranging in [0, 23]), holiday status (1 binary attribute), and power[i] (5 numeric attributes, i.e., $i = 0, -1, \ldots, -4$). The task is to predict power[1] (energy consumption of 1 hour ahead). We have 3500 samples (roughly 5 months) for training and 1000 samples (roughly 1.5 months) for testing.

Result and Discussion Table 3 summarizes quantitative comparisons of FAB/HMEs$^+$, BTLMs and RegTree. With respect to test RMSE, FAB/HMEs$^+$ is slightly better than the others. Further, the number of experts and total complexity of FAB/HMEs$^+$ is much smaller than those of BTLMs and RegTree. Fig. 5 and Table 4 illustrate the learned forecasting model. We

Table 1: Comparison of test RMSEs. The numbers shown in parentheses are standard deviations. The best and second best piecewise linear methods are highlighted in **bold** and ***bold italic*** faces, respectively.

data	sample	dim	FAB/HMEs	FAB/HMEs$^+$	EM/HMEs	BTLMs	RegTree	RBF-SVR
abalone	4177	8	0.80(0.01)	***0.68(0.03)***	0.70(0.03)	**0.67(0.04)**	0.72(0.05)	0.66(0.04)
ailerons	13750	40	0.47(0.03)	**0.41(0.02)**	0.62(0.02)	***0.43(0.01)***	0.46(0.00)	0.40(0.01)
bank32nh	8192	32	0.93(0.05)	**0.68(0.03)**	0.69(0.04)	**0.68(0.04)**	0.78(0.01)	0.67(0.03)
cal-housing	20460	8	0.66(0.05)	***0.56(0.02)***	0.61(0.03)	**0.53(0.03)**	0.64(0.00)	0.48(0.01)
communities	1994	99	0.73(0.06)	0.63(0.04)	**0.60(0.00)**	**0.60(0.03)**	0.65(0.00)	0.59(0.04)
comp-active	8192	22	0.18(0.05)	**0.14(0.01)**	0.54(0.07)	***0.17(0.02)***	0.19(0.02)	0.17(0.02)
kin8nm	8192	8	0.89(0.07)	***0.67(0.03)***	0.77(0.06)	**0.55(0.04)**	0.73(0.03)	0.28(0.01)
pol	15000	48	1.22(0.02)	***0.43(0.14)***	0.77(0.00)	0.59(0.05)	**0.19(0.00)**	0.24(0.01)
puma32H	8192	32	***0.35(0.03)***	**0.33(0.03)**	0.76(0.03)	***0.35(0.17)***	0.36(0.01)	0.46(0.03)

Table 3: Comparison of test RMSE and model complexities (energy demand forecasting).

	FAB/HMEs$^+$	BTLMs	RegTree
RMSE	0.195	0.211	0.197
num of experts	6	11	256
cardinality	4.5	14	1
total	27	154	256

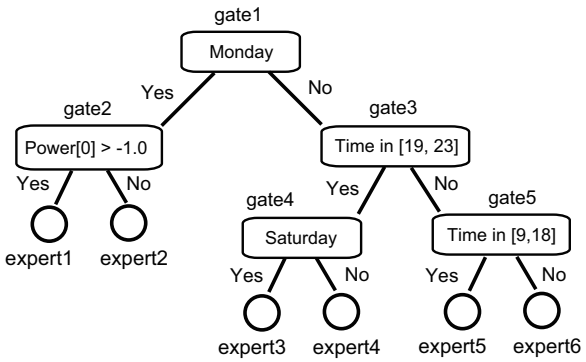

Figure 5: Tree structure of learned model.

can give the following interpretation for this model.

- Since the target is an office building, the energy consumption largely changed between weekday and weekend. FAB/HMEs$^+$ distinguished Monday and Saturday with the others by gate1 and gate4, since they are change points. Expert1, expert2 and expert3 correspond to Monday and Saturday.
- Expert4 and expert6 correspond to time before/after working hour. Since very few people work in this time, the changes of hourly energy consumption are small. Therefore, power[0] is dominant.
- Expert5 is for working hour. Weekend and holidays consume less energy, which appear as negative weight values.

Since there are only 5 gates and experts use roughly 5 features on average, the model interpretation of FAB/HMEs$^+$ is much easier than the others.

Table 4: Coefficients of learned model.

expert	1	2	3	4	5	6
Time	0.17	0.31	0.29	0	0	0
Holiday	-0.16	-0.43	-0.20	0	-0.29	-0.16
Power[0]	1.46	0	0.41	0.83	0.61	0.76
Power[-1]	-1.13	0	0	0	0.05	0
Power[-2]	0.65	0.36	0	0.29	0	0
Power[-3]	-0.19	0	0	0	0	0
Power[-4]	0.65	0	0	-0.19	0	0
Mon	0	0	0	0	0	0
Tue	0	0	0	0.17	0	0.05
Wed	0	0	0	0.07	0	0
Thu	0	0	0	0.14	0	0
Fri	0	0	0	0	0	0
Sat	0	0	0	0	-0.28	0
Sun	0	0	0	0	-0.21	0

5 Summary and Future Work

This paper addressed the model selection issue of PSLMs by extending FAB to HMEs. FAB/HMEs enable us to simultaneously select of a partition structure and local experts' sparsity and offer compact but highly-accurate PSLMs. Also, our new "projection" E-step and "hard-gate" post-processing further improve predictive accuracy by fixing the gap between a predictive objective and FIC maximization. In addition to simulation and benchmark experiments, we confirmed the capability of FAB/HMEs in application to a real-world building energy demand forecasting problem.

A few important studies remain as our future study. While this paper focused on piecewise sparse linear regression, FAB/HMEs as piecewise sparse linear classification are also promising. Technically, the M-step (L_0 regularization for classification) is the most challenging and a FoBa extension for smooth convex loss functions [24, 25] may address the issue. Another important direction is computational efficiency. [16, 17] proposed HME learning algorithms using growing strategies rather than EM-iteration. It is interesting to integrate such ideas into FAB/HMEs learning.

References

[1] Breiman, Leo, Jerome Friedman, R. Olshen and C. Stone. *Classification and Regression Trees.* Wadsworth, 1984. ISBN 0-534-98053-8.

[2] M. Jordan and R. Jacobs. *Hierarchical mixtures of experts and EM algorithm.* Neural Computation, vol. 6, no. 2, pp. 181-214, 1994.

[3] Chipman, H. A., George, E. I., and Mcculloch, R. E. *Bayesian Treed Generalized Linear Models.* In J. M. Bernardo (Ed.), Proceedings Seventh Valencia International Meeting on Bayesian Statistics. Oxford University Press, 2003.

[4] R. Fujimaki and S. Morinaga. *Factorized Asymptotic Bayesian Inference for Mixture Modeling.* In *AISTATS*, 2012.

[5] R. Fujimaki and K. Hayashi. *Factorized Asymptotic Bayesian Hidden Markov Models.* In *ICML*, 2012.

[6] K. Hayashi and R. Fujimaki. *Factorized Asymptotic Bayesian Inference for Latent Feature Models.* In *NIPS*, 2013.

[7] T. Zhang. *Adaptive Forward-Backward Greedy Algorithm for Sparse Learning with Linear Models.* In *NIPS*, 2008.

[8] T. Zhang. *Sparse recovery with orthogonal matching pursuit under RIP.* Information Theory, IEEE Transactions on, 1(1), 17, 2011.

[9] J. Wang and V. Saligrama. *Local supervised learning through space partitioning.* In *NIPS*, 2012.

[10] Z. Xu, M. Kusner, M. Chen and K. Weinberger. *Cost-Sensitive Tree of Classifiers.* In *ICML*, 2013.

[11] A. Frank and A. Asuncin. *UCI machine learning repository.* `http://archive.ics.uci.edu/ml/`, 2010

[12] L. Torgo. *LIACC regression data repository.* `http://www.dcc.fc.up.pt/ ltorgo/Regression/DataSets.html`

[13] S. E. Ryan and L. S. Porth. *A tutorial on the piecewise regression approach applied to bedload transportdata.* Gen. Tech. Rep. RMRS-GTR-189. Fort Collins: US Department of Agriculture, Forest Service, Rocky Mountain Research Station, 2007.

[14] Y. Zhao, R.Schwartz, J. Sroka and J. Makhoul. *Hierarchical Mixtures of Experts Methodology Applied to Continuous Speech Recognition.* In *NIPS*, 1994.

[15] J. R. New and L. E. Parker. *Predicting Future Hourly Residential Electrical Consumption: A Machine Learning Case Study* Energy and Buildings, vol.49, no.0, pp.591-603, 2012.

[16] J. Fritsch, M. Finke and A. Waibel *Adaptively growing hierarchical mixtures of experts.* In *NIPS*, 1997.

[17] S. Waterhouse and A. Robinson. *Constructive algorithms for hierarchical mixtures of experts.* In *NIPS*, 1996.

[18] J. S. Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[19] T. Hofmann, B. Schlkopf, A. J. Smola. *Kernel methods in machine learning.* Annals of Statistics, vol.36, no.2008, pp.1171-1220, 2008.

[20] R. A. Jacobs, M. I. Jordan, S. J. Nowlan,G. E. Hinton. *Adaptive mixtures of local experts.* Neural Computation, vol.3, no.1, pp.79-87, 1991

[21] R. Neal and G. E. Hinton. *A view of the EM algorithm that justifies incremental, sparse, and other variants.* In M. I. Jordan (Ed.), Learning in graphical models. Cambridge, MA: MIT Press, 1999.

[22] S. Watanabe. *Algebraic geometry and statistical learning.* Cambridge University Press, 2009.

[23] A. J. Smola and B. Schlkopf. *A tutorial on support vector regression.* Statistics and Computing , vol.14, no.3, pp.199-222, 2004

[24] A. Jalali, C. Johnson and P. Ravikumar. *On Learning Discrete Graphical Models Using Greedy Methods.* In *NIPS*, 2011.

[25] J. Liu, R. Fujimaki and J. Ye. *Forward-Backward Greedy Algorithms for General Convex Smooth Functions over A Cardinality Constraint.* In *ICML*, 2014.