GROUP-WISE FEATURE SELECTION FOR SUPERVISED LEARNING

Qi Xiao[1], *Hebi Li*[2]. *Jin Tian*[2], *Zhengdao Wang*[1]

[1] Department of ECpE, Iowa State University
[2] Department of Computer Science, Iowa State University

ABSTRACT

Feature selection has been explored in two ways, global feature selection and instance-wise feature selection. Global feature selection picks the same feature selector for the entire dataset, while instance-wise feature selection allows different feature selectors for different data instances. We propose group-wise feature selection, a new setting that sits between global and instance-wise feature selections. In group-wise feature selection, we constrain the number of possible feature selectors to be a finite number K, which allows different feature selectors while regularizing the number of different selectors. This is for flexible trade-offs between expressiveness and model complexity. We propose two techniques to solve the problem: the first applies K-Means Clustering to the instance-wise feature selection algorithm; the second uses the mixture of experts model with Gumbel-Softmax to learn group membership and feature selector simultaneously. We evaluate our techniques and show promising results on both synthetic and real datasets.

Index Terms— Group-wise Feature Selection, Supervised Learning, Deep Neural Networks

1. INTRODUCTION

Feature selection is the technique to perform dimension reduction by selecting relevant features. Feature selection can save computational time and provide better interpretation for both unsupervised and supervised learning scenarios. Global feature selection [1] has been widely explored in the supervised learning literature. Global feature selection aims to find a single binary feature selector for all the data instances with the small cardinality while keeping good prediction accuracy.

Global feature selection is not applicable in cases where important features vary in different examples. For example, in an object recognition task in image processing, relevant features for different images may differ due to the different positions of objects. Instance-wise feature selection [2, 3] are proposed to select different features for different data samples. In [2], each data instance is constrained to have an equal number of relevant features, while [3] relaxes the constraint so that each instance can have any number of relevant features.

We are seeking a trade-off between global feature selection and instance-wise feature selection. We aim to find a set of feature selectors with size K such that all instances' feature selectors are in this set, and we call it group-wise feature selection (GroupFS). We are seeking a trade-off between expressiveness and interpretability. Global feature selection is simple and interpretable but not expressive because it cannot handle the cases where different instances have different takes on relevant features. Instance-wise feature selection is expressive to handle the above cases. However, it loses global interpretability: every instance has a potentially different selector, and it is hard to see the feature importance pattern for the entire data distribution. GroupFS tries to be both expressive and interpretable, by learning K different groups within each constant feature selector.

GroupFS is related to feature selection in Mixture of Experts (MoE) models [4, 5]. The Mixture of Experts model is an ensemble model that divides the input region into several parts and learns an expert model for each region. [6, 7] provide comprehensive literature review for MoE. Feature selection in MoE conducts global feature selection for each expert model. [4] proposes a l_2-penalized maximum-likelihood estimator to select features in MoE. They utilize LASSO and SCAD methods in Expectation-Maximization (EM) to obtain sparse feature importance scores. [5] develops an EM algorithm with coordinate ascent and Expectation-Majorization-Maximization algorithm to generate sparse solutions automatically. Both [4] and [5] focus on linear experts, while GroupFS can fit nonlinear models easily. GroupFS can be trained by end-to-end gradient descent.

We propose two approaches to solve the problem. The first approach is a two-step approach that first trains instance-wise feature selectors and then applies K-Means clustering to the selectors. However, the two steps cannot be optimized simultaneously because K-Means clustering is discontinuous. Our second approach utilizes an MoE model to decide the groupness of data instances. The MoE-based group feature selector can be jointly trained with the discriminator to overcome this difficulty. Evaluation of our models on synthetic and real datasets shows promising performance.

2. PRELIMINARY

2.1. Global Feature Selection

Given size n labeled data $\{(x^i, y^i) : i = 1, \ldots, n\}$ where $x^i \in \mathcal{X} \in \mathbb{R}^d$ and y^i is classification or regression labels, global feature selection aims to find a feature selector $s \in \{0,1\}^d$ such that for almost every $x \in \mathcal{X}$ and selected feature $x_s = x \odot s$, we have

$$\min_s \|s\|_0 \tag{1}$$

$$s.t.\ (Y|X = x) \overset{d.}{=} (Y|X_S = x_s) \tag{2}$$

where d denotes the equality in the distribution. For the selected feature $x_s = x \odot s$, $x = [x_1, ..., x_d] \in \mathcal{X}$ is the input feature vector and $s = [s_1, ..., s_d] \in \{0,1\}^d$ is the binary selection vector, where $s_j = 1$ indicates that the feature x_j is selected; and $s_j = 0$ means x_j is not selected. $x \odot s$ is the element-wise product of x and s that keeps the relevant feature values and set all the irrelevant features to be value 0.

Global feature selection's target is to find a single feature selector s for all the data samples with the smallest number of 1s while keeping the discriminative prediction capacity. In the literature, there are three categories of global feature selection methods:

filters, wrappers, and embedded methods [1]. Filter methods are pre-processing methods not considering output y while wrapper and embedded methods involve the information of y. Wrappers utilize the learning model as a black box to score subsets of features based on their predictive power. Embedded methods perform feature selection during the training process and are usually specific to given learning models.

2.2. Instance-wise Feature Selection

Instance-wise feature selection [2, 3] computes a different selector for each instance and shows better performance than global feature selection. INVASE [3] is an algorithm for instance-wise feature selection. Given size n labeled data $\{(x^i, y^i) : i = 1, \dots, n\}$ where $x^i \in \mathcal{X}$ and y^i can be classification or regression labels, INVASE aims to find a selector function $S_I : \mathcal{X} \rightarrow \{0,1\}^d$ such that for almost every $x \in \mathcal{X}$ and $x_{S_I(x)} = x \odot S_I(x)$ we have

$$(Y|X = x) \overset{d.}{=} (Y|X_{S_I(x)} = x_{S_I(x)}) \tag{3}$$

where $S_I(x)$ is minimal (i.e. fewest 1s) such that equation 3 holds. INVASE defines an objective function based on the KL divergence between the two distributions:

$$L(S_I) = E_x[\int_{\mathcal{Y}} P_Y(y|x)[\log P_Y(y|x) - \log P_Y(y|x_{S_I(x)})]dy + \lambda \|S_I(x)\|_0] \tag{4}$$

where $\|.\|_0$ denotes the l_0 norm and $\lambda \geq 0$ controls the number of selected features. As density function $P_Y(y|x)$ and $P_Y(y|x_{S_I(x)})$ are unknown, INVASE introduces a pair functions f_ϕ and f_γ to estimate $P_Y(y|x)$ and $P_Y(y|x_{S_I(x)})$ respectively. They propose to approximate the selector function S_I by using a single neural network: $\hat{S}^\theta : \mathcal{X} \rightarrow [0,1]^d$ parameterized by weights θ, that outputs a probability for selecting each feature. f_ϕ, f_γ and $\hat{S}^\theta$ are modeled by neural networks, and the parameters of the three networks are updated alternatively using stochastic gradient descent.

2.3. Gumbel-Softmax Estimator

Stochastic neural networks rarely use categorical latent variables due to the inability to back-propagate through samples. Gumbel-Softmax [8, 9] is a re-parametrization trick that replaces the non-differentiable sample from a categorical distribution with a differentiable sample. Let z be a categorical variable with class probabilities $\pi_1, \dots, \pi_K$. Samples of z are k-dimensional one-hot vectors. Samples generator by Gumbel-Softmax trick is:

$$q_i = \frac{\exp(\tau^{-1}(\log(\pi_i) + b_i))}{\sum_{j=1}^{K} \exp(\tau^{-1}(\log(\pi_j) + b_j))}, \tag{5}$$

where $b_i, \dots, b_K$ are i.i.d samples generated from distribution Gumbel(0,1). The Gumbel(0,1) [10] distribution can be sampled by first drawing a sample u_i from Uniform(0,1) and computing $b_i = -\log(-\log(u_i))$.

2.3.1. *Straight-Through Gumbel-Softmax Estimator*

For scenarios where we need discrete samples, q can be discretized using argmax, but the continuous approximation will be used in the back-propagation by approximating $\nabla_z \approx \nabla_q$.

3. PROPOSED METHODS

3.1. Problem Formulation

We propose a Group-wise Feature Selection (GroupFS) problem. The goal is to find feature selector mapping for $S_G : \mathcal{X} \rightarrow F$ and a set of feature selectors $F = \{s^k | k = 1, ..., K, s^k \in \{0,1\}^d\}$ such that for almost all $x \in \mathcal{X}$, we have

$$(Y|X = x) \overset{d.}{=} (Y|X_{S_G(x)} = x_{S_G(x)}). \tag{6}$$

3.2. INVASE with K-Means clustering

In this section, we will illustrate how to obtain GroupFS based on instance-wise feature selection. One natural idea is to apply a clustering algorithm on the instance-wise feature selector. We propose a method based on the clustering algorithm. The procedure is applying the clustering algorithm after we obtain the instance-wise feature selector. It has two steps.

1. First, train an instance-wise feature selector. Each data sample has an individual feature selector.
2. Apply the K-means clustering [11, 12] to all the feature selectors. Each sample's feature selector will be assigned a cluster membership. The center for its assigned cluster is the obtained group-wise feature selector. We will threshold cluster centers value by 0.5 to obtain binary vectors.

3.3. GroupFS with Mixture of Experts Selector

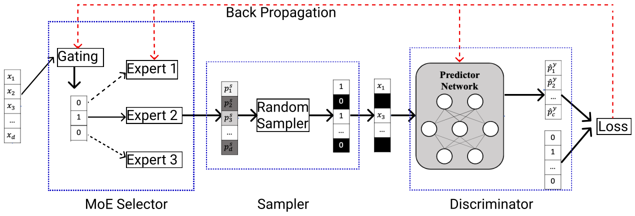

Fig. 1: GroupFS-MoE Model Architecture

Alternatively, we present a neural network based architecture designed for group-wise feature selection. We use feature importance score $w \in [0,1]^d$ to quantity the probability of a feature being selected. Feature selector s can be sampled from w, and w is treated as a continuous approximation for s. To constrain the value w to $[0,1]^d$ in deep neural networks, we use a re-parametrization trick for w. Let $w = \text{sigmoid}(v) = \frac{1}{1+\exp(-v)} \in (0,1)$. Given importance score w, we assume feature selector s follows the multivariate Bernoulli distribution with parameter w,

$$\pi(s; w) = \prod_{i=1}^{d} w_i^{s_i}(1 - w_i)^{(1-s_i)} \tag{7}$$

Assume the feature importance score for all the data samples belongs to the set $\{w^1, ..., w^K\}$. For input x, we assume it will choose only one of the feature importance scores. If w^k is picked as feature importance for x, all the other scores will not be considered. We will use a gating network $g(x; \theta)$ to represent the exclusive membership. $g(x; \theta)$ is an one-hot vector satisfying

$$\sum_{k=1}^{K} g_k(x; \theta) = 1, g_k(x; \theta) \in \{0,1\}, \tag{8}$$

where $g_k(x;\theta)$ is the k-th dimension of the gating model $g(x;\theta)$ output.

The Group-wise Feature Selection can be divided into two steps:

1. First the input data will be injected into a gating network $g(x;\theta)$
2. If $g_k(x;\theta) = 1, g_{i\neq k}(x;\theta) = 0$, x will pick the feature importance score w^k

Thus, we model the feature importance score for input x:

$$\pi(s|x;\theta,w^1,...,w^k) = \textstyle\sum_{k=1}^{K} g_k(x;\theta)\pi_k(s;w^k), \tag{9}$$

where

$$\pi_k(s;w^k) = \textstyle\prod_i^d (w^k)_i^{s_i}(1-(w^k)_i)^{1-s_i} \tag{10}$$

This kind of structure is a Mixture of Experts (MoE) model with discrete gating. $g(x;\theta)$ outputs a categorical vector; thus gradient descent cannot be applied to train θ. Straight-Through Gumbal-Softmax trick is applied to $g(x;\theta)$ as a continuous approximation such that gradient descent can used. The gating function has the following formulation:

$$g_k(x;\theta) = \frac{\exp(\tau^{-1}(\log(o_k)+b_k))}{\sum_{j=1}^{K}\exp(\tau^{-1}(\log(o_j)+b_j))}, \tag{11}$$

where o_k is the original output of gating for k-th feature selector. When calculating the output prediction, argmax will be applied to g to obtain discrete gating, while equation 11 will be used for calculating gradient.

Through the gating function, each data sample is assigned an feature importance score $w_k, k \in \{1,\dots,K\}$. Then the feature selector s is sampled from $\pi(x;w_k)$ and the supervised learning model will output $f(x,s;\phi)$ as prediction result, where ϕ is the parameters of model f. The group-wise feature selection model architecture is drawn in figure 1.

The loss for the data pair (x,y) is written as:

$$l(y,x,s;\phi) = -\textstyle\sum_{i=1}^{c}[y_i\log f(x,s;\phi) - y_i\log f'(x;\gamma)] \tag{12}$$

where c is the number of classes and f' is the discriminative model learning from the entire feature with parameter γ. We call f' as the baseline model.

Assuming feature selector s is sparse, we will add l_0 norm for s. The loss can be written as:

$$\begin{aligned}
&L(\theta,w^1,...,w^k)\\
=&\mathbf{E}_{(X,Y)\sim P}\mathbf{E}_{s\sim\pi(s|x)}[l(y,x,s;\phi)+\lambda||s||_0]\\
=&\mathbf{E}_{(X,Y)\sim P}[\textstyle\sum_{s\in\{0,1\}^d}\sum_{k=1}^{K} g_k(x;\theta)\pi_k(s;w^k)l(y,x,s;\phi)]\\
&+\lambda\textstyle\sum_{i=1}^{d}\sum_{k=1}^{K} g_k(x;\theta)(w^k)_i,
\end{aligned} \tag{13}$$

where $\pi(s|x)$ is short for $\pi(s|x;\theta,w^1,...,w^k)$ for simplicity. We use gradient descent to optimize the $\theta,w^1,...,w^k,\phi,\gamma$ as in INVASE.

4. EXPERIMENTS

4.1. Evaluation on Synthetic Data

4.1.1. Performance evaluation

Normalized Mutual Information (NMI) is used to measure the performance. In synthetic datasets, we have the ground truth group labels. NMI can measure the consistency between the predicted group label and the ground truth one. Assume the two group label assignments are two random variables C_1 and C_2, and the entropy of C_1 and C_2 are $\mathrm{H}(C_1)$ and $\mathrm{H}(C_2)$. The Mutual Information between them is written as $\mathrm{I}(C_1;C_2)$. The Normalized Mutual Information is defined as:

$$\mathrm{NMI}(C_1;C_2) = \frac{2\mathrm{I}(C_1;C_2)}{\mathrm{H}(C_1)+\mathrm{H}(C_2)}. \tag{14}$$

The score will be at $[0,1]$, where 0 indicates independence and 1 means maximal dependence. Mutual Information is symmetric, thus NMI is symmetric:

$$\mathrm{NMI}(C_1;C_2) = \mathrm{NMI}(C_2;C_1) \tag{15}$$

4.1.2. Synthetic dataset

In this section, we show results on synthetic classification datasets. We adopt the same synthetic datasets generation procedure presented in INVASE. Each data sample is i.i.d generated from standard normal distribution. Specifically, the feature $x \in \mathbb{R}^{11}$, and the output y is generated according to $P(y=1|x) = \frac{1}{1+\mathrm{logit}(x)}$, where $\mathrm{logit}(x)$ can be following three functions:

$$\mathrm{Syn1}(x) = \begin{cases}\exp(x_1x_2), & x_{11}<0\\ \exp(\sum\limits_{i=3}^{6} x_i^2 - 4), & \text{otherwise}\end{cases} \tag{16}$$

$$\mathrm{Syn2}(x) = \begin{cases}\exp(x_1x_2), & x_{11}<0\\ \exp(-10\sin 2x_7 + 2\|x_8\| + x_9 + \exp(-x_{10})), & \text{otherwise}\end{cases} \tag{17}$$

$$\mathrm{Syn3}(x) = \begin{cases}\exp(\sum\limits_{i=3}^{6} x_i^2 - 4), & x_{11}<0\\ \exp(-10\sin 2x_7 + 2\|x_8\| + x_9 + \exp(-x_{10})), & \text{otherwise}\end{cases} \tag{18}$$

4.1.3. Models

We evaluate two approaches and set $K = 2$ for all the synthetic datasets. The model INVASE+KM is the two-step approach that first generates instance-wise feature selectors for each instance and subsequentially runs the K-Means clustering algorithm on the feature instances. The GroupFS-MoE approach uses the Mixture of Experts model to produce Group-wise feature selectors. For the model INVASE+KM, the model for INVASE part is the same as in the paper [3]. For our GroupFS-MoE model, we differ from INVASE only in the selector part.

4.1.4. Results

Table 1: Evaluation of proposed methods on synthetic datasets

	Syn1			Syn2			Syn3		
	NMI	MSE	Acc	NMI	MSE	Acc	NMI	MSE	Acc
INVASE+KM	.828	.189	.703	.904	.178	.715	.925	.136	.810
GroupFS-MoE	**.911**	.182	.710	**.921**	.177	.715	**.960**	.131	.811

In Table 1, we show the NMI, Mean Squared Error (MSE), and discriminator accuracy (Acc) for the three models on three sets of synthetic data. As we can see, the 2-step approach INVASE+KM

Table 2: Learned GroupFS Feature Selectors for Syn1,Syn2,Syn3

	Syn1			Syn2			Syn3		
Experts	E1	E2	g	E1	E2	g	E1	E2	g
#samples	1617	1717		1621	1713		1621	1713	
x1	1	-	-	1	-	-	.02	.01	-
x2	1	-	-	1	-	-		.02	-
x3	-	1	-	.02	.01	-	1	-	-
x4	-	1	-	.01	.01	-	1	-	-
x5	-	1	-	.01	.01	-	1	-	-
x6	-	1	-	.02	.01	.01	1	-	-
x7	.02	-	-	-	1	-	-	1	-
x8	.03	.01	-	-	.99	-	-	1	-
x9	.03	-	-	-	1	-	-	1	-
x10	.03	-	-	-	1	-	-	1	-
x11	.17	1	.21	.02	.05	.27	.99	.99	.33

and the GroupFS-MoE performs well. The NMI value is more than 90%, indicating that the methods consistently recovered the correct grouping.

Compared with the 2-step INVASE+KM approach, GroupFS-MoE performs better in clustering consistency, showing that jointly training the grouping and feature selectors yield better grouping results. Note that the discriminator of INVASE+KM and GroupFS-MoE achieves similar accuracy, which indicates that both models can train the discriminator well, but GroupFS-MoE can generate better grouping results thanks to simultaneous training.

The result of feature selectors of the GroupFS-MoE model is shown in Table 2. The purpose is to show the correctness of the expert feature selectors and the gating network weights qualitatively. The #samples row shows how many samples are gated to different experts by the gating network. About an equal amount of samples are in each cluster, which is consistent with our synthetic data generating procedure that assigns clustering by the sign of x_{11}. The rows (x_1 to x_{11}) show the feature selectors and gating weights for all 11 input features. For each dataset, the columns E1 and E2 show the feature selector for each expert, and the column g shows the corresponding weight in the gating network for each input feature. Compared with synthetic data ground truth, we can see that the expert networks learned by the GroupFS-MoE model produce the true feature selectors for the data instances: for example, in Syn1, x_1, x_2 are the features used for group 1, and x_3, x_4, x_5, x_6 are used for group 2. In all datasets, x_{11} is used to decide the groups. The gating network correctly learns this: the learned gating network has non-zero weight only at x_{11}. The result also shows that x_{11} may be classified as the relevant feature.

4.2. Real Data Experiments

4.2.1. Dataset and Evaluation Metrics

We test our algorithm on two real regression datasets, Boston housing and baseball Salary. Boston housing is from StatLib archive [13]. For Boston housing, the task is to predict house prices from the provided 13 factors, such as crime rate, housing age. Baseball Salary dataset is available on the website of Journal of Statistics Education. The data contains the year 1992 salaries along with 16 performance measures for 337 major league baseball players. The task is to analyze how the performance measures affect salaries.

For both datasets, we consider grouping the feature selectors into 2 clusters. Both tasks are regression problems, and no ground truth clustering labels are available; thus, we cannot compute the mutual information and compare it with baseline clustering methods. Therefore in this evaluation, we evaluate our approach in terms of the regression mean squared error. For the discriminator model and baseline model, we use two hidden layers with dimension d.

4.2.2. Models in Comparison

We compare our model with Regularized MoE (RMoE) models [4, 14, 5] because RMoE models also generate group-wise feature selectors. Similar to our GroupFS-MoE model, RMoE models use a gating network to determine the groupness of data instances; unlike our model, they train a separate discriminator for each group and apply regularization for each discriminator. Such separate discriminator design makes their model more complex and challenging to fit. Specifically, the model in [4] is a LASSO regularized MoE model. The MIXLASSO in [14] is a LASSO regularized mixture model without a gating network. The two papers [4, 14] are from the same Author Abbas Khalili, thus we refer to the two models as Khalili's model. The LASSO+l_2 model of [5] is a MoE model with mixed LASSO and l_2 regularizers. We also report the vanilla MoE with no feature selection that is used in LASSO+l_2 paper.

Table 3: Discriminator's Mean Square Error (MSE) comparison with Regularized MoE. The Khalili [4] and LASSO+l_2 [5] models are only evaluated on training data, thus we compare our model's training data MSE with them. We also show our model's performance on testing data where we hold a 80%-20% split with showing average of 12 runs with different random seeds. INV+KM is short for INVASE+KMeans. GroupFS is short for GroupFS-MoE

	Training				Testing	
	Khalili[4]	lasso+l_2[5]	GroupFS	INV+KM	GroupFS	INV+KM
Boston	.2044	.1989	.0879	.0853	.1863	0.1846
Baseball	1.1858	.2821	.2371	.2480	.3056	.3417

4.2.3. Performance analysis

The quantitative mean square errors are shown in Table 3. Since Khalili and LASSO+l_2 are only evaluated on training data, we report and compare the performance of our model on the training data as well. We also supply our model's performance on testing results. For both datasets, GroupFS-MoE and INVASE+KMeans has lower training MSE. This shows that the Mixture of Experts discriminator model is complex to fit, the MoE-based discriminator has to learn two separate expert models, while we only use MoE for feature selection, and the discriminator is one single neural network model same as INVASE, and can fit better. Our testing error on Boston data is even lower than Khalili and LASSO+l_2's training error.

5. CONCLUSION

In this paper, we proposed a new feature selection schema called group-wise feature selection that sits between global feature selection and instance-wise feature selection. We propose two approaches to solve the problem, including INVASE+KMeans and an MoE-based feature selector. We evaluated our model on synthetic and real datasets and showed better performance than existing regularized MoE models on real datasets.

6. REFERENCES

[1] Manoranjan Dash and Huan Liu, "Feature selection for classification," *Intelligent data analysis*, vol. 1, no. 1-4, pp. 131–156, 1997.

[2] Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan, "Learning to explain: An information-theoretic perspective on model interpretation," in *International Conference on Machine Learning*. PMLR, 2018, pp. 883–892.

[3] Jinsung Yoon, James Jordon, and Mihaela van der Schaar, "Invase: Instance-wise variable selection using neural networks," in *International Conference on Learning Representations*, 2018.

[4] Abbas Khalili, "New estimation and feature selection methods in mixture-of-experts models," *Canadian Journal of Statistics*, vol. 38, no. 4, pp. 519–539, 2010.

[5] Faicel Chamroukhi and Bao-Tuyen Huynh, "Regularized maximum likelihood estimation and feature selection in mixtures-of-experts models," *Journal de la société française de statistique*, vol. 160, no. 1, pp. 57–85, 2019.

[6] Seniha Esen Yuksel, Joseph N Wilson, and Paul D Gader, "Twenty years of mixture of experts," *IEEE transactions on neural networks and learning systems*, vol. 23, no. 8, pp. 1177–1193, 2012.

[7] Saeed Masoudnia and Reza Ebrahimpour, "Mixture of experts: a literature survey," *Artificial Intelligence Review*, vol. 42, no. 2, pp. 275–293, 2014.

[8] Eric Jang, Shixiang Gu, and Ben Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[9] Chris J Maddison, Andriy Mnih, and Yee Whye Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *arXiv preprint arXiv:1611.00712*, 2016.

[10] Emil Julius Gumbel, *Statistical theory of extreme values and some practical applications: a series of lectures*, vol. 33, US Government Printing Office, 1954.

[11] Stuart Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[12] James MacQueen et al., "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Oakland, CA, USA, 1967, vol. 1, pp. 281–297.

[13] "Boston data," `http://lib.stat.cmu.edu/datasets/boston`, 2010, Accessed: 2010-09-30.

[14] Abbas Khalili and Jiahua Chen, "Variable selection in finite mixture of regression models," *Journal of the american Statistical association*, vol. 102, no. 479, pp. 1025–1038, 2007.