2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Mixture Modeling and Inference for Recognition of Multiple Recurring Unknown Patterns

Zeyu You[1], Raviv Raich[1], and Yonghong Huang[2]
[1]School of Electrical Engineering and Computer Science
Oregon State University Corvallis, Oregon 97331-5501
[2]Intel Labs, 2111 Northeast 25th Avenue, Hillsboro, Oregon 97124
{youz@onid.orst.edu, raich@eecs.oregonstate.edu, catherine.huang@intel.com}

***Abstract*— We consider the problem of finding unknown patterns that are recurring across multiple sets. For example, finding multiple objects that are present in multiple images or a short DNA code that is repeated across multiple DNA sequences. Earlier work on the topic includes a statistical modeling approach in which the same template is placed at a random position in multiple independent sets. Using mixture modeling, we propose an extension to the approach that allows the detection of multiple templates placed across multiple sets. Moreover, we present an expectation-maximization algorithm for jointly estimating multiple templates based on a mixture of non-Gaussian distributions. To address the non-convexity of the problem, a robust initialization method is presented and theoretical guarantees are provided. We evaluate the performance of the algorithm on both synthetic data and real-world data consisting of electrical voltage recordings of home appliance activations. Our results indicate that the proposed algorithm significantly improves the detection accuracy relative to the single pattern model.**

I. Introduction

Finding recurring patterns in data can be applied to various areas, such as finding regulatory sequences in DNA [1], pattern matching in strings [2], and audio motif discovery for bioacoustic applications [3].

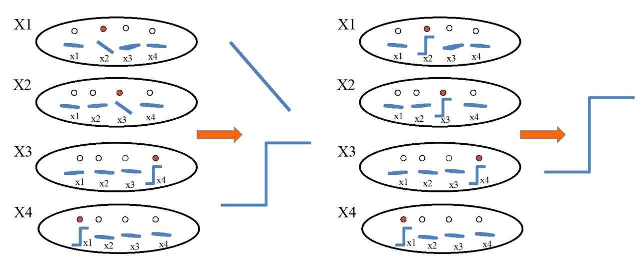
Different approaches have been proposed for a pre-specified pattern matching. A Gibbs sampling framework for estimating and identifying multiple patterns in the DNA sequences is proposed in [1], while a graph based WINNOWER algorithm for finding a signal in sampled DNA sequence is proposed in [4]. In computer science, fast pattern matching [2] for text strings has been widely used. Dynamic time warping (DTW) is also a well-known algorithm for a matching problem that allows variations in time [5]. If the pattern of interest is unknown, the problem becomes a blind pattern recognition problem. In [6], a parameter-free CK distance approach with probabilistic early abandoning is proposed for audio motif discovering on large data archives. Finding the most similar pair in long sequence is their focus.

A natural extension to the single pattern matching involves the recognition of multiple recurring patterns. For multiple motif identification and alignment of protein sequences, [7] proposes a combination of search and refinement algorithm. For speaker identification [8], a robust text-independent Gaussian mixture model is proposed.

We introduce a novel non-Gaussian mixture model based on the single pattern model in [9]. Due to the non-convex nature of the problem, multiple local solutions may arise. To address this problem, we propose novel robust initialization and iterative updates. Based on mixture modeling approach, we first show estimation performance on synthetic data. Then, we present detection performance on real world dataset and show a significant increase in performance compared to the approaches of [9] and [10].

II. Background and Related Work

Before we delve into the problem of estimating multiple different templates from N multi-instance bags containing only one of the multiple templates (see Fig. 1(a)), we start by introducing the simpler problem of estimating a single template from N multi-instance bags each containing only one occurrence of the desired template (see Fig. 1(b)). In Fig. 1(a) and (b), the dot over the template indicates the position of template in the bag.



(a) Multiple templates (b) Single template

Fig. 1. Recognition of templates in multiple sets.

A. Single Pattern Model

We consider the setting in which a single pattern is embedded exactly once in each of N bags. Each bag contains multiple instances. We denote the number of instances in the ith bag by n_i. We denote the ith bag by $\mathcal{X}_i$ and the jth-instance in this bag by $\mathbf{x}_{ij}$. Consequently, we can write the ith bag as $\mathcal{X}_i = \{\mathbf{x}_{i1}, \ldots, \mathbf{x}_{in_i}\}$. One can consider two different tasks. The first involves estimating the desired pattern and the second involves determining the position of the desired pattern in each bag (see Fig. 1(b)).

In [9], a statistical model for this setting is developed to offer means for conducting a performance analysis. Specifically, the Cramér–Rao lower bound (CRLB) on the mean-squared-error (MSE) of the estimator of the desired pattern is evaluated against the number of bags, the number of instances in each bag, the signal-to-noise ratio, and the dimension of the signal. Our goal in this paper is to develop estimation algorithms to learn the hidden patterns for the single pattern model in [9] and to extend both model and algorithms to the multiple template case. We begin by reviewing the single pattern model.

To describe the generative model of the dataset, we begin by assuming that the dataset consists of N independent bags $\{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_N\}$. To generate the ith bag, first a position random variable (RV) J_i is generated uniformly in $\{1, 2, \ldots, n_i\}$, i.e., $P(j_i = j) = 1/n_i$. Then, the instances $\mathbf{x}_{ij}$ are generated such that for $j \neq j_i$ $\mathbf{x}_{ij} \sim \mathcal{N}(0, \sigma^2 I_d)$ and for $j = j_i$ $\mathbf{x}_{ij} \sim \mathcal{N}(\mathbf{s}, \sigma^2 I_d)$. In other words, a noisy version of the template $\mathbf{s}$ is embedded in the position j_i and zero-mean Gaussian noise is placed in all other positions. Consequently, the probability density function (PDF) for a single bag is given by $\sum_j P(j)\big(\prod_{j'=1\neq j}^{n_i} \mathcal{N}(\mathbf{x}_{ij'}; 0, \sigma^2 I_d)\big)\mathcal{N}(\mathbf{x}_{ij}; \mathbf{s}, \sigma^2 I_d)$. Simplifying this expression yields the following PDF of a single bag:

$$G_i(X_i|\mathbf{s}) = \prod_{j'=1}^{n_i} \frac{1}{\sqrt{2\pi\sigma^2}^d} e^{-\frac{\|\mathbf{x}_{ij'}\|^2}{2\sigma^2}} \frac{1}{n_i} \sum_{j=1}^{n_i} e^{\frac{2\mathbf{s}^T \mathbf{x}_{ij} - \|\mathbf{s}\|^2}{2\sigma^2}}. \quad (1)$$

The log-likelihood for N independent bags is

$$\sum_{i=1}^{N} \log G_i(X_i|\mathbf{s}) = C - \frac{N\|\mathbf{s}\|^2}{2\sigma^2} + \sum_{i=1}^{N} \log\Big(\sum_{j=1}^{n_i} e^{\frac{\mathbf{s}^T \mathbf{x}_{ij}}{\sigma^2}}\Big), \quad (2)$$

where C denotes terms which are independent of the parameter $\mathbf{s}$. To learn the unknown template $\mathbf{s}$, we proceed with maximum-likelihood (ML) solution. Note that this model is a mixture of models which vary in the position of the unknown template $\mathbf{s}$. Consequently, the ML problem is non-convex. We propose an initialization approach with quantitative guarantees and refinements to solve the ML estimation.

1) Initialization: In [9], an iterative method is provided for estimating $\mathbf{s}$. However, it is pointed out that the results depend on the initialization. In this paper, we introduce a core idea which suggests that despite the non-convex nature of the problem, a close to optimal solution can be obtained. We rely on the observation that the log-likelihood can be approximated using the soft-max approximation of the max function: $\log(\sum_i e^{\alpha_i}) \approx \max_i \alpha_i$, yielding,

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^{N} \log G_i(X_i|\mathbf{s}) &\approx C - \tfrac{\|\mathbf{s}\|^2}{2\sigma^2} + \tfrac{1}{N}\textstyle\sum_{i=1}^{N} \max_j \frac{\mathbf{s}^T \mathbf{x}_{ij}}{\sigma^2} \\ &= \max_{j_1,\ldots,j_N} C - \tfrac{\|\mathbf{s}\|^2}{2\sigma^2} + \frac{\mathbf{s}^T \frac{1}{N}\sum_i \mathbf{x}_{ij_i}}{\sigma^2}. \quad (3)\end{aligned}$$

Consequently, ML can be approximated by

$$\max_{\mathbf{s},\mathbf{j}} -\frac{\|\mathbf{s}\|^2}{2} + \mathbf{s}^T \frac{1}{N} \sum_i \mathbf{x}_{\mathbf{ij_i}}, \quad (4)$$

or as a minimization problem

$$\min_{\mathbf{s},\mathbf{j}} \sum_{i=1}^{N} \|\mathbf{x}_{ij_i} - \mathbf{s}\|^2 - \sum_{i=1}^{N} \|\mathbf{x}_{ij_i}\|^2, \quad (5)$$

where $\mathbf{j} = [j_1, j_2, \ldots, j_N]^T$. This problem is a non-trivial integer programming. A solution to a more general form is proposed in [10]:

$$\min_{\mathbf{s},\mathbf{j}} \sum_{i=1}^{N} \|\mathbf{x}_{ij_i} - \mathbf{s}\|^2 + \sum_{i=1}^{N} \phi_i(\mathbf{x}_{ij_i}), \quad (6)$$

where $\phi_i(\mathbf{x}_{ij_i}) \geq 0$. Minimizing the objective in (6) with respect to $\mathbf{s}$ results in $\mathbf{s} = \frac{1}{N}\sum_{i=1}^{N} x_{ij_i}$. After substituting $\mathbf{s}$ back into (6), a minimization problem only with respect to $\mathbf{j}$ is obtained:

$$\hat{\mathbf{j}} = \arg\min_{\mathbf{j}} f(\mathbf{j}), \qquad \text{where,}$$

$$f(\mathbf{j}) = \tfrac{1}{2N} \sum_{i_1=1}^{N} \sum_{i_2=1}^{N} \|\mathbf{x}_{i_1 j_{i_1}} - \mathbf{x}_{i_2 j_{i_2}}\|^2 + \sum_{i=1}^{N} \phi_i(\mathbf{x}_{ij_i}). \quad (7)$$

The objective in (7) can be viewed as a sum of edge weight in a graph given by $D_{i_1 i_2} = \|\mathbf{x}_{i_1 j_{i_1}} - \mathbf{x}_{i_2 j_{i_2}}\|^2$ and a sum of node penalties $\phi_i(\mathbf{x}_{ij_i})$. The graph is a complete graph since the sum runs over all pairs of (i_1, i_2). The solution for the complete graph requires a brute-force search which results in computational complexity $\mathcal{O}(M^N)$, where M is the number of instances per bag. To reduce the computational complexity, the proposed algorithm in [10] replaces the single complete graph by N bipartite graphs (see Fig. 2), reducing the computational complexity to $\mathcal{O}(M^2N^2)$ [11]. For each bipartite graph, we set aside the ith bag and calculate the sum of the squared distances from one instance in bag i to the other instance in all other bags as a function of $f_i(\mathbf{j}_i)$. Instead of minimizing the objective in (7), a sub-optimal solution $\tilde{\mathbf{j}}$ is obtained by solving N independent minimizations. For each i, we solve

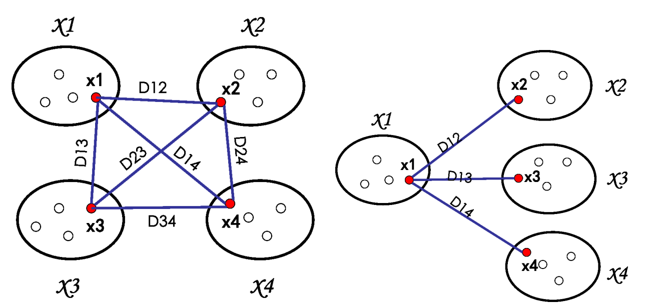


Fig. 2. Graphical representation of two approach: (7) and (9). This figure is a reproduction of the figure from [10].

$$\mathbf{j}^i \quad = \quad \arg\min_{\mathbf{j}} f_i(\mathbf{j}), \;\; \text{where} \quad (8)$$

$$f_i(\mathbf{j}) \quad = \quad \sum_{i_2=1\neq i}^{N} \big(\|\mathbf{x}_{ij_i} - \mathbf{x}_{i_2 j_{i_2}}\|^2 + \phi_{i_2}(\mathbf{x}_{i_2 j_{i_2}})\big). \quad (9)$$

Then, the vector of position estimate is determined by $\tilde{\mathbf{j}} = \mathbf{j}^{i^*}$, where

$$i^* = \arg\min_i f_i(\mathbf{j}^i). \quad (10)$$

In [10], it is shown that the minimum of the objective $f(\mathbf{j})$ can be bounded using the $f_i(\mathbf{j})$'s as follows:

$$\frac{1}{2}\min_i f_i(\mathbf{j}^i) \leq f(\hat{\mathbf{j}}) \leq f(\tilde{\mathbf{j}}) \leq \min_i f_i(\mathbf{j}^i).$$

This sandwich inequality guarantees $f(\hat{\mathbf{j}}) \leq f(\tilde{\mathbf{j}}) \leq 2f(\hat{\mathbf{j}})$. Consequently, the bound suggests that the bi-partite approach yields a solution which guarantees that $f(\tilde{\mathbf{j}})$ the objective value in (7) evaluated at the sub-optimal solution is no more than the twice of its global minimum $f(\hat{\mathbf{j}})$.

Naturally this approach can be applied to the minimization in (5) by setting $\phi_i(\mathbf{x}_{ij_i}) = \max_t \|\mathbf{x}_{it}\|^2 - \|\mathbf{x}_{ij_i}\|^2$ in (6). Consequently the minimum of the objective $\sum_{i=1}^N \|\mathbf{x}_{ij_i} - \mathbf{s}\|^2 + \sum_{i=1}^N (\max_j \|\mathbf{x}_{ij}\|^2 - \|\mathbf{x}_{ij_i}\|^2)$ can be approached within a factor of 2. Moreover, this result suggests that the approximate solution $\mathbf{s}^*$

$$\mathbf{s}^* = \frac{1}{N}\sum_{i=1}^N \mathbf{x}_{i\tilde{\mathbf{j}}} \tag{11}$$

can offer a feasible robust initialization to iterative methods for solving the ML in (2).

2) Refinement: To refine the solution to the nearest optimum of (2), we consider a minimization problem of the negative objective:

$$\begin{aligned} \min_{\mathbf{s}} \quad f(\mathbf{s}) &= u(\mathbf{s}) - v(\mathbf{s}), \text{where,} \\ u(\mathbf{s}) &= \frac{\|\mathbf{s}\|^2}{2\sigma^2}; \\ v(\mathbf{s}) &= \frac{1}{N}\sum_{i=1}^N \log\Big(\sum_{j=1}^{n_i} e^{\frac{\mathbf{s}^T\mathbf{x}_{ij}}{\sigma^2}}\Big). \end{aligned}$$

Since $u(\mathbf{s})$ and $v(\mathbf{s})$ are both real-valued convex functions, $f(\mathbf{s})$ is a convex-concave function and may contain multiple local solutions. We propose majorization-minimization (MM) approach [12]. The general idea is to construct a majorizing function $g(\mathbf{s}, \mathbf{s}^{(t)})$ such that (i) $g(\mathbf{s}, \mathbf{s}^{(t)}) \geq f(\mathbf{s})$ for any $\mathbf{s}, \mathbf{s}^{(t)}$; and (ii) $g(\mathbf{s}, \mathbf{s}^{(t)}) = f(\mathbf{s})$ for any $\mathbf{s}$. Minimizing $g(\mathbf{s}, \mathbf{s}^{(t)})$ function instead of $f(\mathbf{s})$ results in the following update rule $\mathbf{s}^{(t+1)} = \arg\min_{\mathbf{s}} g(\mathbf{s}, \mathbf{s}^{(t)})$, which yields non increasing sequence of the objective, i.e., $f(\mathbf{s}^{(t+1)}) \leq f(\mathbf{s}^{(t)})$.

A simple upper bound function $g(\mathbf{s}, \mathbf{s}^{(t)})$ can be obtained by linearizing the convex function $v(\mathbf{s})$. Since $v(\mathbf{s}) \geq v(\mathbf{s}^{(t)}) + (\mathbf{s} - \mathbf{s}^{(t)})^T \Delta v(\mathbf{s}^{(t)})$, then $f(\mathbf{s}) \leq u(\mathbf{s}) - v(\mathbf{s}^{(t)}) - (\mathbf{s} - \mathbf{s}^{(t)})^T \Delta v(\mathbf{s}^{(t)}) := g(\mathbf{s}, \mathbf{s}^{(t)})$ [12]. Therefore, the upper bound $g(\mathbf{s}, \mathbf{s}^{(t)})$ is:

$$\begin{aligned} g(\mathbf{s}, \mathbf{s}^{(t)}) &= \frac{\|\mathbf{s}\|^2}{2\sigma^2} - \frac{1}{N}\sum_{i=1}^N \cdot \frac{\sum_{j=1}^{n_i} e^{\frac{\mathbf{s}^{(t)T}\mathbf{x}_{ij}}{\sigma^2}} \cdot \frac{x_{ij}}{\sigma^2}}{\sum_{j=1}^{n_i} e^{\frac{\mathbf{s}^{(t)T}\mathbf{x}_{ij}}{\sigma^2}}} \\ &\quad \cdot \big(\mathbf{s} - \mathbf{s}^{(t)}\big) - v(\mathbf{s}^{(t)}). \end{aligned}$$

By minimizing $g(\mathbf{s}, \mathbf{s}^{(t)})$ with respect to $\mathbf{s}$, we obtain the update rule:

$$\mathbf{s}^{(t+1)} = \frac{1}{N}\sum_{i=1}^N \sum_{j=1}^{n_i} \frac{e^{\frac{\mathbf{s}^{(t)T}\mathbf{x}_{ij}}{\sigma^2}}}{\sum_{k=1}^{n_i} e^{\frac{\mathbf{s}^{(t)T}\mathbf{x}_{ik}}{\sigma^2}}} \mathbf{x}_{ij}. \tag{12}$$

In effort to obtain the global solution, we propose the combination of the initialization in (11) and the iterations in (12). Inspired by this approach for solving the ML problem for the single template case, we proceed with a mixture model generalization for the multiple template case.

III. Problem Formulation

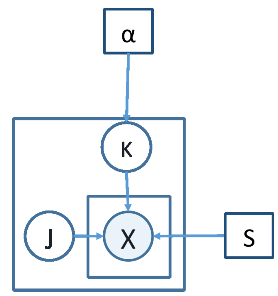


Fig. 3. A graphical model for the K-Pattern alignment problem

To formulate this problem, consider N subsets $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_N$ of the d-dimensional Euclidean space $\mathbb{R}^d$, i.e., $\mathcal{X}_i \subseteq \mathbb{R}^d$ for $i = 1, 2, \ldots, N$. Each set is assumed to contain only one of K possible patterns $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_K\}$ among other instances (see Fig. 1(a)). Our goal is to obtain the K patterns of interest.

A. Statistical K-pattern Model

To model the problem of finding the K-unknown elements in multiple sets in a noisy setting, we extend the single pattern model in [9] as shown in Fig. 3. We introduce hidden template id RV K in addition to the position of the template J in a given bag.

For each bag i, we organize the elements of $\mathcal{X}_i$ in a $d \times n_i$ matrix $X_i = [\mathbf{x}_{i1}, \ldots, \mathbf{x}_{in_i}]$ and consider joint distribution of the observations represented by the observation matrix $X = [X_1, \ldots, X_N]$ given the unknown vectors $\mathbf{s}_1, \ldots, \mathbf{s}_K$. We introduce the class prior probability α_k that satisfies $0 < \alpha_k < 1, \sum_{k=1}^K \alpha_k = 1$ for each probability density function $G(X_i|\mathbf{s}_k)$ in (1). Since we assume that sets are generated in an independent fashion, we express the joint distribution of sets as a product of their marginal PDFs:

$$\Lambda(X;\theta) = \prod_{i=1}^N f_i(X_i;\theta) \tag{13}$$

$$f_i(X_i;\theta) = \sum_{k=1}^K \alpha_k G(X_i|\mathbf{s}_k), \tag{14}$$

where $G(X_i|\mathbf{s}_k)$ is a the ith bag probability density function conditioned on template pattern $\mathbf{s}_k$, and $\theta = \{\alpha_1, \alpha_2, \ldots, \alpha_K, \mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_K\}$. Then, the log-likelihood function is:

$$\log\Lambda(X;\theta) = \sum_{i=1}^N \log\Big(\sum_{k=1}^K \alpha_k G(X_i|\mathbf{s}_k)\Big). \tag{15}$$

Although the expectation maximization algorithm has been well-developed to solve the parameter estimation problem in mixture models, the optimization of a non-convex objective is non-trivial.

IV. Inference Solution Framework

In the following section, we present the expectation maximization (EM) framework for solving the parameter estimation problem in general. Furthermore, we use a Majorization-minimization (MM) approach with robust initialization for implementing the EM updates.

A. Expectation Maximization

EM is an iterative solution to maximum likelihood [13]. Specifically, the iterations offer a non-decreasing sequence of the likelihood function. In general, the auxiliary function $Q(\theta, \theta^{(t)})$ is:

$$\begin{aligned} Q(\theta, \theta^{(t)}) = \quad & E[\log P(X_1, X_2, \dots, X_N, k_1, k_2, \dots, k_N; \theta) \\ & |X_1, X_2, \dots, X_N, \theta^{(t)}] \end{aligned}$$

The iterations are performed in two steps. In the E-step, the auxiliary function is computed as:

$$Q(\theta, \theta^{(t)}) = \sum_{i=1}^{N} \sum_{k=1}^{K} p_i^{(t)}(k|\theta^{(t)}) \log(\alpha_k G(X_i|\mathbf{s}_k)).$$

Here, $p_i^{(t)}(k|\theta^{(t)}) = \frac{\alpha_k^{(t)} G(X_i|\mathbf{s}_k^{(t)})}{\sum_{l=1}^{K} \alpha_l^{(t)} G(X_i|\mathbf{s}_l^{(t)})}$ represents the probability that the ith bag was generated by component K.
In the M-step, we maximize the auxiliary function $\max_\theta \; Q(\theta, \theta^{(t)})$ to obtain the update rule:

$$\begin{aligned} \alpha_k^{(t+1)} &= \frac{1}{N} \sum_{i=1}^{N} p_i^{(t)}(k|\theta^{(t)}), & (16) \\ \mathbf{s}_k^{(t+1)} &= \arg\max_{\mathbf{s}_k} \quad \sum_{i=1}^{N} p_i^{(t)}(k|\theta^{(t)}) \cdot \\ & \qquad \Big(C - \frac{\|\mathbf{s}_k\|^2}{2\sigma^2} + \log\big(\sum_{j=1}^{n_i} e^{\frac{\mathbf{s}_k^T \mathbf{x}_{ij}}{\sigma^2}}\big)\Big). & (17) \end{aligned}$$

The optimization in (17) involves the sum of convex-concave functions that cannot be solved in closed-form. We propose to solve (17) and obtain $\mathbf{s}_k^{(t+1)}$ by using a method described in Section II. First, we find a robust initialization for $\mathbf{s}_k^{(t+1)}$ (i.e., $\mathbf{s}_k^{(t+1,0)}$). Then we use MM approach to refine the solution.

Algorithm 1 Expectation Maximization for mixtures of G likelihood function

Initialize $\theta^0 = \{\alpha_1^0, \alpha_2^0, \dots, \alpha_K^0, s_1^0, s_2^0, \dots, s_K^0\}$.
procedure EMforMGF(θ^0, X)
  while Likelihood $\Lambda(X; \theta)$ not converged **do**
    E-step: compute membership probability $p_{ik}^{(t+1)} = \frac{\alpha_k^{(t)} G(x_i|s_k^{(t)})}{\sum_{l=1}^{K} \alpha_l^{(t)} G(x_i|s_l^{(t)})}$
    M-step: $\max_\theta \quad Q(\theta, \theta^{(t)})$ to obtain s_k
    Running Procedure: $\hat{s_k}$ =MMforS(s_k^0, X)
  Return θ

B. Robust Initialization

There are two sets of initialization parameters $\alpha_k^0 = \{\alpha_1^0, \alpha_2^0, \dots, \alpha_K^0\}$ and $\mathbf{s}_k^0 = \{\mathbf{s}_1^0, \mathbf{s}_2^0, \dots, \mathbf{s}_K^0\}$. The initialization of Gaussian mixture model is a well-known problem (e.g., see [14]). We can directly apply initialization techniques for the α_k^0 and $\mathbf{s}_k^0$, while initializing $\mathbf{s}_k^{(t+1,0)}$ is our focus.

By approximating the log of sum of exponential functions with the largest term in the sum $\log(\sum_{j=1}^{n_i} e^{\mathbf{s}_k^T \mathbf{x}_{ij}}) \approx \max_j \; \mathbf{s}_k^T \mathbf{x}_{ij}$ and $p_{ik} = p_i(k|\theta)$, $w_{ik} = \frac{p_{ik}}{\sum_{i=1}^{N} p_{ik}}$, the approximated maximization problem in (17) becomes:

$$\begin{aligned} \max_{\mathbf{s}_k} \qquad & \sum_{i=1}^{N} w_{ik} \cdot \big(-\frac{\|\mathbf{s}_k\|^2}{2} + \max_{j_i} \quad \mathbf{s}_k^T \mathbf{x}_{ij_i}\big), \quad \text{or}, \\ \max_{\mathbf{s}_k, \mathbf{j}} \qquad & \sum_{i=1}^{N} w_{ik} \cdot \big(-\frac{\|\mathbf{s}_k\|^2}{2} + \quad \mathbf{s}_k^T \mathbf{x}_{ij_i}\big). \qquad (18) \end{aligned}$$

We first solve for $\mathbf{s}_k$ by taking the derivative of the objective function with respect to $\mathbf{s}_k$ and setting to zero. We obtain the solution for $\mathbf{s}_k$ as $\mathbf{s}_k = \sum_{i=1}^{N} w_{ik} \mathbf{x}_{ij}$. Substituting $\mathbf{s}_k$ back into (18), yields:

$$\begin{aligned} \max_{\mathbf{j}} \qquad & \frac{1}{2}\big(\sum_{i=1}^{N} w_{ik} \mathbf{x}_{ij_i}\big)^2 \quad \text{or}, \\ \max_{\mathbf{j}} \qquad & \frac{1}{2} \sum_{i_1=1}^{N} \sum_{i_2=1}^{N} w_{i_1 k} w_{i_2 k} \mathbf{x}_{i_1 j_{i_1}}^T \mathbf{x}_{i_2 j_{i_2}}, \end{aligned}$$

which can be written as,

$$\min_{\mathbf{j}} \quad f^{(k)}(\mathbf{j}), \qquad \text{where}$$

$$\begin{aligned} f^{(k)}(\mathbf{j}) \quad &= \tfrac{1}{2} \sum_{i_1=1}^{N} \sum_{i_2=1}^{N} w_{i_1 k} w_{i_2 k} ||\mathbf{x}_{i_1 j_{i_1}} - \mathbf{x}_{i_2 j_{i_2}}||^2 \\ & + \sum_{i_1=1}^{N} w_{i_1 k} \big(\max_t \quad ||\mathbf{x}_{i_1 t}||^2 - ||\mathbf{x}_{i_1 j_{i_1}}||^2\big). \quad (19) \end{aligned}$$

The objective in (19) can be viewed as a weighted sum of edge weight in a graph given by $D_{i_1 i_2} = \|\mathbf{x}_{i_1 j_{i_1}} - \mathbf{x}_{i_2 j_{i_2}}\|^2$ and a weighted sum of node penalties $\phi_i(j) = \max_t \; ||\mathbf{x}_{it}||^2 - ||\mathbf{x}_{ij}||^2$.

This problem is similar to the single pattern matching problem. We apply the bi-partite graph approach for each pattern to robustly initialize $\mathbf{s}_k^{(t)}$ for each iteration with estimated $\hat{\mathbf{s}}_k$. Since (19) is similar to (7), we can use the same procedure to obtain the ML solution $\hat{\mathbf{j}}_k$. Using $f_i^{(k)}(\mathbf{j})$ functions and solving N minimizations for each pattern individually, we obtain the approximate solution $\tilde{\mathbf{j}}_k$:

$$\begin{aligned} \mathbf{j}_k^{i_1} \quad &= \quad \arg\min_{\mathbf{j}} f_{i_1}^{(k)}(\mathbf{j}_{i_2}), \;\; \text{where} \\ f_{i_1}^{(k)}(\mathbf{j}_{i_2}) \quad &= \quad \sum_{i_2=1 \neq i_1}^{N} w_{i_2 k}\big(\|\mathbf{x}_{i_1 j_{i_1}} - \mathbf{x}_{i_2 j_{i_2}}\|^2 + \phi_{i_2}(j_{i_2})\big). \end{aligned}$$

Then, $\tilde{\mathbf{j}}_k = \mathbf{j}_k^{i_k^*}$, where

$$i_k^* = \arg\min_i f_{i_1}^{(k)}(\mathbf{j}_{i_2}).$$

Based on the approximate solution $\tilde{\mathbf{j}}_k$, we directly obtain the approximate estimation for $\mathbf{s}_k^*$:

$$\mathbf{s}_k^* = \sum_{i=1}^{N} w_{ik} \mathbf{x}_{i\tilde{\mathbf{j}}_k}. \tag{20}$$

Moreover, we can still establish a lower and upper bound for each pattern k:

$$\frac{1}{2} \sum_{i_1} w_{i_1 k} f_{i_2}^{(k)}(\mathbf{j}_k^{i_1}) \leq f^{(k)}(\tilde{\mathbf{j}}_{(k)}) \leq \min_{i_1} f_{i_1}^{(k)}(\mathbf{j}_k^{i_1}).$$

Since $\sum_{i_1} w_{i_1 k} \min_{i_1} f_{i_1}^{(k)}(\mathbf{j}_k^{i_1}) \leq \sum_{i_1} w_{i_1 k} f_{i_1}^{(k)}(\mathbf{j}_k^{i_1})$, we can further bound the lower bounded by $\frac{1}{2} \min_{i_1} f_{i_1}^{(k)}(\mathbf{j}_k^{i_1})$. Therefore,

$$\frac{1}{2} \min_{i_1} f_{i_1}^{(k)}(\mathbf{j}_k^{i_1}) \leq f^{(k)}(\hat{\mathbf{j}}_k) \leq f^{(k)}(\tilde{\mathbf{j}}_k) \leq \min_{i_1} f_{i_1}^{(k)}(\mathbf{j}_k^{i_1}).$$

This bound shows that the robust initialization finds out an approximated template such that the corresponding objective is within a factor of 2 from the optimal solution objective.

Algorithm 2 Robust Initialization

Input p_{ik} from previous E-step in EM algorithm
Compute $w_{ik} = \frac{p_{ik}}{\sum_{i=1}^N p_{ik}}$
procedure SEARCHGOODINSTANCES(w_{ik}, X)
    for bagid i_1 in 1,...,N **do**
        for bagid i_2 in 1,...,N$\neq i_1$ **do**
            Compute weighted distance matrix $D_{j_{i_1} j_{i_2}} = w_{i_2 k}\big(\|\mathbf{x}_{i_1 j_{i_1}} - \mathbf{x}_{i_2 j_{i_2}}\|^2 + \phi_{i_2}(j_{i_2})\big)$
            Find smallest instance position for each i_1: $[j_1^*, j_2^*, \ldots, j_N^*]$=minindex($D^T_{j_{i_1} j_{i_2}}$)
            Compute $v = v + D^T_{j_{i_1} j_{i_2}}$
        Find overall smallest distance value for each i_1:
        MinVal($\mathrm{i_1}$)=minimum value(v)
        MinIdx($\mathrm{i_1}$)=minimum index(v)
    [i_1^*]=min(MinVal($\mathrm{i_1}$))
    Get $[j_1*, j_2*, \ldots \mathrm{MinIdx}(\mathrm{i}_1^*), \ldots, j_N*]$ from optimal position collection in bag i_1^*.
    Return $\mathbf{s}_k = \sum_{i=1}^N w_{ik} \mathbf{x}_{ij}$

C. Majorization-minimization for ML refinement

In the M-step of the EM algorithm, a separate update rule is used for each $\mathbf{s}_k$ (see (18)). We can directly apply MM algorithm for each individual minimization:

$$\min_{\mathbf{s}_k} \quad \tilde{f}_k(\mathbf{s}_k), \qquad \text{where}$$

$$\tilde{f}_k(\mathbf{s}_k) = \frac{\|\mathbf{s}_k\|^2}{2\sigma^2} - \sum_{i=1}^{N} w_{ik} \log\Big(\sum_{j=1}^{n_i} e^{\frac{\mathbf{s}_k^T \mathbf{x}_{ij}}{\sigma^2}}\Big),$$

where $p_{ik} = p_i(k|\theta)$ and $w_{ik} = \frac{p_{ik}}{\sum_{i=1}^N p_{ik}}$. The upper bound of the objective $g_k(\mathbf{s}_k, \mathbf{s}_k^{(t')})$ is a majorizing function which satisfies $\tilde{f}_k(\mathbf{s}_k) \leq g_k(\mathbf{s}_k, \mathbf{s}_k^{(t')})$. By minimizing g_k function, a solution of $\mathbf{s}_k*$ is obtained in the t'th iteration and it provides an input to the $(t'+1)$th iteration:

$$g_k(\mathbf{s}_k, \mathbf{s}_k^{(t')}) = \frac{\|\mathbf{s}_k\|^2}{2\sigma^2} - \sum_{i=1}^N w_{ik} \frac{\sum_{j=1}^{n_i} e^{\frac{\mathbf{s}_k^{(t')T}\mathbf{x}_{ij}}{\sigma^2}} \cdot \frac{\mathbf{x}_{ij}}{\sigma^2}}{\sum_{j=1}^{n_i} e^{\frac{\mathbf{s}_k^{(t')T}\mathbf{x}_{ij}}{\sigma^2}}} \cdot(\mathbf{s}_k - \mathbf{s}_k^{(t')}) - \sum_{i=1}^N w_{ik} \log\Big(\sum_{j=1}^{n_i} e^{\frac{\mathbf{s}_k^{(t')T}\mathbf{x}_{ij}}{\sigma^2}}\Big).$$

Then by setting $\frac{\delta g_k(\mathbf{s}_k, \mathbf{s}_k^{(t')})}{\delta \mathbf{s}} = 0$, we have the update rule:

$$\mathbf{s}_k^{(t'+1)} = \sum_{i=1}^N w_{ik}^{(t'+1)} \sum_{j=1}^{n_i} W_{ijk}^{(t'+1)} \cdot \mathbf{x}_{ij}, \quad \text{where,}$$

$$w_{ik}^{(t'+1)} = \frac{p_{ik}^{(t')}}{\sum_{i=1}^N p_{ik}^{(t')}}, \; W_{ijk}^{(t'+1)} = \frac{e^{\frac{\mathbf{s}_k^{(t')T}\mathbf{x}_{ij}}{\sigma^2}}}{\sum_{j=1}^{n_i} e^{\frac{\mathbf{s}_k^{(t')T}\mathbf{x}_{ij}}{\sigma^2}}}. \tag{21}$$

Using a combination of robust initialization and iterative implementation of the ML estimator of $\mathbf{s}_k$, we can obtain the solution of $\mathbf{s}_k^{(t+1)}$ in (17).

Algorithm 3 Majorization-minimization for template $\mathbf{s}_k$

RobustInitialize $s\mathbf{s}_k^0 = \{\mathbf{s}_1^0, \mathbf{s}_2^0, \ldots, \mathbf{s}_K^0\}$.
procedure MMFORS($\mathbf{s}_k^0, X$)
    while Likelihood $f(\mathbf{s}; \mathbf{s}^{(t')})$ not converged **do**
        Recalculate $Q_{ik}^{(t'+1)} = \frac{p_{ik}^{(t')}}{\sum_{i=1}^N p_{ik}^{(t')}}$ from E-step of EM
        Recalculate $W_{ijk}^{(t'+1)} = \frac{e^{\frac{\mathbf{s}_k^{(t')T}\mathbf{x}_{ij}}{\sigma^2}}}{\sum_{j=1}^{n_i} e^{\frac{\mathbf{s}_k^{(t')T}\mathbf{x}_{ij}}{\sigma^2}}}$
        Update $\mathbf{s}_k^{(t'+1)} = \sum_{i=1}^N Q_{ik}^{(t'+1)} \sum_{j=1}^{n_i} W_{ijk}^{(t'+1)} \cdot \mathbf{x}_{ij}$
    Return $\mathbf{s}_k^{final}$

V. PERFORMANCE ANALYSIS

In this section, we evaluate our proposed method on a synthetic data set and on a real world data set of electric appliance activations (*Source: Pecan Street Research Institute*). We evaluate our methods in terms of Receiver Operating Characteristic curve (ROC) and Area Under the ROC curve (AUC) and compare the results to the results presented in [10]. We also show the improved performance on ROC and AUC based on the mixture model.

A. Synthetic Dataset Generation

The X_i's are generated in an independent fashion based on the K-pattern Model, where the template id for bag i is uniformly sampled in $\{1, 2, \ldots, K\}$ and the template position in the ith bag J_i is uniformly sampled in $\{1, 2, \ldots, M\}$. We choose $K = 3$ and ground-truth templates $\mathbf{s}_1(t), \mathbf{s}_2(t), \mathbf{s}_3(t)$ are designed as:

$$\begin{aligned} \mathbf{s}_1(t) &= u(t + D/2) - u(t - D/2), \quad \text{for} \quad t = 0, 1, 2, \ldots, D; \\ \mathbf{s}_2(t) &= t, \quad \text{for} \quad 0 \leq t \leq D; \\ \mathbf{s}_3(t) &= -t, \quad \text{for} \quad 0 \leq t \leq D. \end{aligned}$$

Note that $u(t)$ is a step function. We normalize each vector $\mathbf{s}_i = [\mathbf{s_i(1)}, \mathbf{s_i(2)}, \dots, \mathbf{s_i(t)}]^\mathbf{T}$ using $\mathbf{s}_i/||\mathbf{s}_i||$ and set it as our new $\mathbf{s}_i$ for all $i = 1, 2, 3$.

Since the estimated accuracy is affected by the set of parameters $\{D, M, N, \mathrm{SNR}\}$ (D-dimension of the template, M-number of instances per bag, N-number of bags and SNR-signal to noise ratio), we perform numerical experiments to analyze the mean squared error (MSE) of the iterative implementation of the ML estimator against different setups of parameters. Then, we also perform a detection task based on a maximum a-posterior probability (MAP) detector using the estimated patterns.

B. Estimation Performance Evaluation on Synthetic Dataset

To analyze the estimation performance with respect to different parameters, we start with the nominal setting of $N = 50$ sets with $M = 20$ $D = 100$-dimensional elements in each set for SNR $\in \{-20\mathrm{dB}, -18\mathrm{dB}, \dots, 20\mathrm{dB}\}$. Then we vary one parameter (D, M, and N) at a time as $D \in \{100, 400\}$ and $M, N \in \{10, 50\}$ to evaluate the MSE of the ML estimator as a function of SNR. For each combination of parameters $\{N, M, D, \mathrm{SNR}\}$, we generate 50 independent Monte-Carlo (MC) realizations based on our mixture model. Since EM is sensitive to the initialization, we use 10 iterations of different random values of α_k^0 and $\mathbf{s}_k^0$ and choose the estimate yielding the largest likelihood value. Using the 50 MC runs, we compute the sum of each k empirical MSE with the mean and its confidence interval. In Fig. 4, we present the MSE as a function of the SNR of the iterative implementation of the ML estimator. Increasing SNR and the number of bags N yields a decrease in the relative MSE, while increasing template dimension D and the number of instances in each bag M yields a small increase in the relative MSE when SNR is less than $-10dB$. We also notice that it is possible to achieve an under $-10dB$ relative MSE, for fairly low values of SNR by either increasing the dimension D or the number of sets N. This suggests that using more sets compensate for the performance degradation when choosing larger number of elements in each set.

C. Detection Performance Evaluation on Synthetic Dataset

We designed a GLRT framework for detecting the position $\mathbf{J}$ of unknown patterns $\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_K\}$ given a new dataset $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\}$. We denote $\mathbf{x}_j*$ as an instance in the set that contains one of the true templates $\mathbf{s}_k \in \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_K\}$. Our goal is to detect the position j in each bag and analyze the performance of our detectors as a function of k.

By maximizing the posterior probability of $\mathbf{J}$ and $\mathbf{K}$, which can be written as $P(J = j, K = k|X) = \frac{f(X|J=j,K=k)\cdot P(J=j)P(K=k)}{\sum_{j=1}^M \sum_{k=1}^K f(X|J=j,K=k)\cdot P(J=j)P(K=k)} \propto f(X|J = j, K = k)P(J = j)P(K = k)$, we can directly obtain the detector as

$$\max_{J,K} \; P(J = j, K = k|X).$$

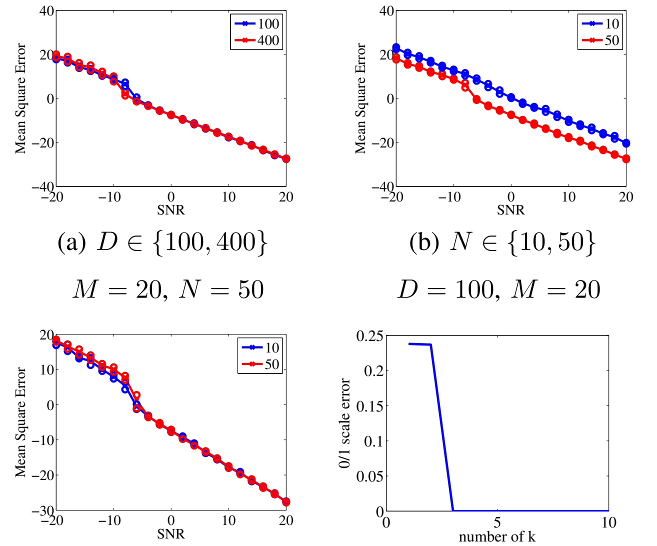
To simplify the notation, we omit the dependence of $P(J = j, K = k|X; \alpha, \mathbf{s})$ on α and $\mathbf{s}$ and write it as $P(J = j, K = k|X)$. Since for each bag, $f(X_i|J = j, K = k) = \prod_{j=1}^M f_0(x_j) \cdot \frac{f_1(x_j|s_k)}{f_0(x_j)}$, and based on the Gaussian model for f_1 and f_0, the log of the posterior probability can be rewritten as:

$$\begin{aligned} \log(f(X|J = j, K = k)) &= -\frac{||x_j - s_k||^2}{2\sigma^2} + \frac{||x_j||^2}{2\sigma^2} + C; \\ \log(P(J = j)) &= -\log(M); \\ \log(P(K = k)) &= \log(\alpha_k). \end{aligned}$$

By taking the negative $\log P(J = j, K = k|X)$, we obtain the detector as:

$$\min_{j,k} \; \frac{2s_k^T x_j - ||s_k||^2}{2\sigma^2} + \log(\alpha_k). \tag{22}$$

In this experiment, we apply this detector to the synthetic data set with 50 bags and we detect the position of the pattern based on the K-pattern estimation results of $\hat{\mathbf{s}}_k, \hat{\alpha}_k$. If the position of a pattern is true, we count it as a hit, otherwise, we count it as a miss. The error given by $P(J \neq j|X)$ is presented in Fig. 4(d) as a function of the number of the templates.



(a) $D \in \{100, 400\}$ $M = 20$, $N = 50$

(b) $N \in \{10, 50\}$ $D = 100$, $M = 20$

(c) $M \in \{10, 50\}$ $D = 100$, $N = 50$

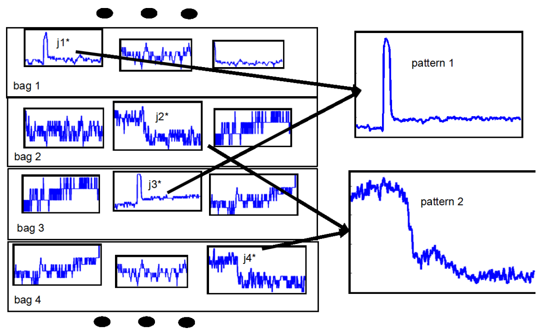
(d) Detection error vs. K

Fig. 4. MSE of the ML estimator as a function of SNR in (i)-(iii) and detection error vs. K in (iv) .

D. Real World Dataset

We use the same dataset as described in [10]. The Pecan Street dataset contains four homes (PS025, PS029, PS046, PS051) voltage peak to peak measurements and power submeter measurements. In [10], the dataset is divided into training data within the period 11/17/2012-11/25/2012 and test data within the period 11/26/2012-12/11/2012. Moreover, it is shown that the blind joint delay estimation for single activation pattern yields mean AUC around 75%. However, for some appliances such as oven, the AUC is as low as

50%. Our goal is jointly identify multiple activation patterns together for the same appliance in a multiple bag setting. We show an example of oven (in Fig. 5) with two activation patterns repeated multiple times.



(e) Oven Activations

Fig. 5. Examples of Identifying Two Activation Patterns of Oven.

To make the comparison fair, we use the same amount of training examples and choose a window of size 700 (i.e., $D = 700$) during the training phase. We compare the ROC and AUC in [10] with those of the K-pattern model for both single activation appliances and multi-activation appliances.

E. Detection Performance Evaluation on Real-World Dataset

Since not all appliances have multiple activation patterns, we test the performance of our proposed algorithm by increasing K (the number of patterns). Based on the activation patterns estimated during the training phase, we apply the same detector $\max_\tau \sum_{t=1}^{T}(y_{test}(t) - \bar{y}_{test}(t))(s(t-\tau) - \overline{s(t-\tau)}) \underset{H_0}{\overset{H_1}{\gtrless}} \rho''$ as in [10] to each hourly file in the test data with a period of more than ten days and acquire the ROC curve for each appliance in each of the four homes. In [10], because the model is not robust to outliers, the training data has been filtered. To make the comparison fair, we also apply the filtering process to the training data such that the training examples are free from outliers. The corresponding AUCs for all appliances available in each home on both single pattern model and K-pattern model is present in the TABLE I.

We observe that mixture model with $K = 1$ attains a reasonable ROC curves and the similar AUCs compared to the single pattern model in [10]. We also notice that AUCs for most appliances in the mixture model is slightly higher than the single pattern model (10/14 entries are higher in the mixture model in Table I). This suggests that the mixture model, built upon the single pattern model, is not decreasing the performance of the single pattern model. Moreover, we concentrate on increasing the AUCs for the appliances revealing more than one activation pattern.

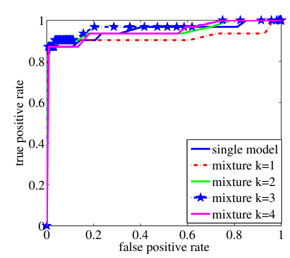
By increasing the number of activation patterns K in the model, more than one activation pattern can be identified, but each pattern would be more coarse. This is due to the effect of averaging with less bags for each pattern. To capture the variation among patterns and maintain the completeness of the training data, we train the mixture model on the unfiltered training data. Then, we apply the same detector to test for different K.

House ID	App. Name	AUC (Single)	AUC (Mixture$K = 1$)
PS025	Air-Cond.	0.95066	0.95228
PS025	Oven	0.52177	0.52076
PS029	Air-Cond.	0.91496	0.91496
PS029	Fridge	0.71906	0.68795
PS029	Furnace	0.86338	0.86519
PS029	Dryer	0.99142	0.98460
PS029	Microwave	0.87869	0.87926
PS029	Oven	0.91030	0.91602
PS046	Air-Cond.	0.84892	0.85882
PS046	Fridge	0.49252	0.49680
PS046	Furnace	0.53887	0.57652
PS046	Oven	0.91824	0.95471
PS051	Air-Cond.	0.91311	0.91418
PS051	Oven	0.78501	0.77505

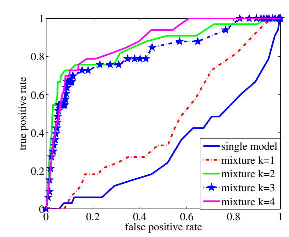
TABLE I

AUCs OF SINGLE PATTERN MODEL VS. MIXTURE MODEL WITH $K = 1$ ON TEST DATASET

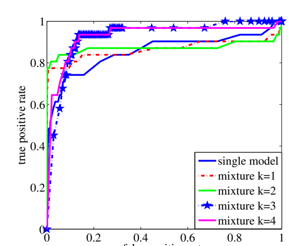
For appliances with only one activation pattern, such as air-conditioning, furnace, and microwave, considering a larger K model would not affect the performance significantly (see Fig. 6(a) and (c)). For those appliances with multiple activation patterns, such as oven, dryer and fridge, mixture model captures more variations of the activation patterns yielding a significant improvement in detection accuracy (see Fig. 6(b) and (d)).



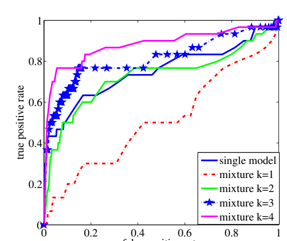
(a) PS025 Air ROC



(b) PS046 Oven ROC



(c) PS029 Furnace ROC



(d) PS046 Fridge ROC

Fig. 6. ROC plots for Single pattern detection and for multiple pattern detection.

In the case of $K = 1$, the performance of single pattern model and mixture model is similar (see TABLE I). To test the performance of mixture model by the effect of varying K, we present the AUCs for $K = 1$, $K = 2$, $K = 3$ and

$K = 4$ in four homes which is shown in TABLE II. Even though increasing the number K in the training phase is more time consuming, the detection accuracy has increased in the testing phase. The computation complexity of training the mixture model with K components is K times more than the single pattern model. However, the AUCs improved significantly for $K = 3$ than the single pattern model, especially for those appliances containing multiple activation patterns (see Table I and Table II). Moreover, without manually filtering the training data, we can save time and avoid human intervention.

House ID	App. Name	AUC ($K = 1$)	AUC ($K = 2$)	AUC ($K = 3$)	AUC ($K = 4$)
PS025	Air-Cond.	0.91536	0.94718	0.96430	0.94521
PS025	Oven	0.62589	0.77490	0.77117	0.78919
PS029	Air-Cond.	0.89135	0.93373	0.93373	0.93373
PS029	Fridge	0.69454	0.82033	0.82255	0.77734
PS029	Furnace	0.92872	0.87298	0.92531	0.92872
PS029	Dryer	0.14849	0.98840	0.96926	0.97028
PS029	Microwave	0.70571	0.94661	0.93492	0.92364
PS029	Oven	0.92116	0.95478	0.95478	0.91151
PS046	Air-Cond.	0.27775	0.85115	0.94300	0.95887
PS046	Fridge	0.50963	0.72579	0.81084	0.87933
PS046	Furnace	0.50576	0.55790	0.57826	0.52459
PS046	Oven	0.45611	0.85403	0.81768	0.87384
PS051	Air-Cond.	0.91362	0.93661	0.96314	0.96314
PS051	Oven	0.77862	0.78036	0.75660	0.79800

TABLE II
AUCS OF MIXTURE MODEL BY VARYING K

We observe that the AUCs for most appliances change slightly when varying K from 1 to 4, while some appliances change significantly, such as oven in home PS025, fridge, dryer and microwave in home PS029 and air-conditioning, fridge in home PS046. We notice that the AUC may not increase by increasing K because outliers can be recognized as a pattern introducing more 'false alarms'. Even if some appliances have only one activation pattern, the mixture model approach increases the AUC by capturing the variations in patterns. In practice, K can be selected using cross-validation to prevent overfitting.

VI. Discussion and Conclusion

In this paper, we introduced a statistical mixture model for finding multiple patterns across multiple sets. We provided an EM-based inference framework with robust initialization approach. We tested the performance of our proposed algorithms on both synthetic dataset and real world dataset. The results on synthetic data showed that for high SNR, MSE for multiple patterns would achieve a similar performance as that of the single pattern. In real world dataset, for some appliances, we observed a significant performance increase when using the K pattern model instead of the single pattern model. Moreover, if a home appliance has only one activation pattern, using the mixture model maintained the performance of the single pattern model. The mixture model introduces significant performance improvement relative to the single pattern model when an appliance exhibits multiple activation patterns.

VII. Acknowledgement

We would like to thank Intel corporation for partially funding this research and extend a special thanks to Tomm Aldridge and James Song of Intel Labs. We would also like to thank the Pecan Street Research Institute and Intel corporation for providing the dataset used in this paper.

References

[1] Charles E Lawrence, Stephen F Altschul, Mark S Boguski, Jun S Liu, Andrew F Neuwald, and John C Wootton, "Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment," *science*, vol. 262, no. 5131, pp. 208–214, 1993.

[2] Donald E Knuth, James H Morris, Jr, and Vaughan R Pratt, "Fast pattern matching in strings," *SIAM journal on computing*, vol. 6, no. 2, pp. 323–350, 1977.

[3] Abdullah Mueen, Eamonn J Keogh, Qiang Zhu, Sydney Cash, and M Brandon Westover, "Exact discovery of time series motifs," in *SIAM International Conference on Data Mining (SDM)*. SIAM, 2009, pp. 473–484.

[4] Pavel A Pevzner, Sing-Hoi Sze, et al., "Combinatorial approaches to finding subtle signals in dna sequences," in *ISMB*, 2000, vol. 8, pp. 269–278.

[5] Donald J Berndt and James Clifford, "Using dynamic time warping to find patterns in time series," in *KDD workshop*. Seattle, WA, 1994, vol. 10, pp. 359–370.

[6] Yuan Hao, Mohammad Shokoohi-Yekta, George Papageorgiou, and Eamonn Keogh, "Parameter-free audio motif discovery in large data archives," in *IEEE International Conference on Data Mining (ICDM)*. 2013, IEEE.

[7] Timothy L Bailey and Charles Elkan, "Fitting a mixture model by expectation maximization to discover motifs in bipolymers," 1994.

[8] Douglas A Reynolds and Richard C Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 1, pp. 72–83, 1995.

[9] Raviv Raich and You Zeyu, "Looking for the same needle in multiple haystacks: Performance bounds," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, accepted.

[10] You Zeyu, Raviv Raich, and Huang Yonghong, "An inference framework for detection of home appliance activation from voltage measurements," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, accepted.

[11] John Adrian Bondy and U. S. R. Murty, *Graph Theory with Applications*, North-Holland, 1976.

[12] Gert R Lanckriet and Bharath K Sriperumbudur, "On the convergence of the concave-convex procedure," in *Advances in neural information processing systems*, 2009, pp. 1759–1767.

[13] Todd K Moon, "The expectation-maximization algorithm," *Signal processing magazine, IEEE*, vol. 13, no. 6, pp. 47–60, 1996.

[14] Usama M Fayyad, Cory Reina, and Paul S Bradley, "Initialization of iterative refinement clustering algorithms," in *KDD*, 1998, pp. 194–198.