L0-regularized Sparsity for Mixture Models

Dzung Phan and Tsuyoshi Ide
IBM Research

SIAM International Conference on Data Mining (SDM19)
Outline

▪ Problem setting

▪ Mixture weight update formulation

▪ Quadratic time algorithm

▪ Experimental results
Outline

- Problem setting
- Mixture weight update formulation
- Quadratic time algorithm
- Experimental results
A Gaussian mixture model (GMM) is a weighted sum of $K$ component Gaussian densities

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x | \mu_k, \Sigma_k)$$

- $\pi_k$: mixture weights
- $(\mu_k, \Sigma_k)$: the mean and covariance

Irrelevant components can be mistakenly included in the training model
Sparse Mixture of Sparse Gaussian Graphical Model

- A Gaussian mixture model (GMM) is a weighted sum of \( K \) component Gaussian densities

\[
p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k)
\]

- Irrelevant components can be removed by a sparse model

\[
\forall k: \pi_k \geq 0 \quad \sum_{k=1}^{K} \pi_k = 1
\]

- \( \pi_k \): sparse mixture weights
- \( \sum_{k}^{-1} \): sparse inverse covariance

Irrelevant components can be removed by a sparse model
Expectation-Maximization (EM) Algorithm

- We suggest a penalized log-likelihood as

\[
\log \mathcal{L}_P(\theta) = \sum_{n=1}^{N} \log \left( \sum_{k=1}^{K} \pi_k N(x_n | \mu_k, \Sigma_k) \right) - \lambda \sum_{k=1}^{K} \phi(\pi_k, \Sigma_k)
\]

- EM Algorithm
  
  - **E-step**: Evaluate the responsibilities (posterior probability of data point \( i \) belonging to mixture component \( k \))
  - **M-step**: Use the updated responsibilities to re-estimate the parameters \( \theta = (\pi_k, \mu_k, \Sigma_k) \)

- Updating mixture weights is as follows if \( \phi \equiv 0 \)

\[
\max_{\pi} \sum_{k=1}^{K} r_k \ln \pi_k \quad \text{subject to} \quad \sum_{k=1}^{K} \pi_k = 1
\]
Expectation-Maximization (EM) Algorithm

- We suggest a penalized log-likelihood as

\[
\log \mathcal{L}_P(\theta) = \sum_{n=1}^{N} \log \left( \sum_{k=1}^{K} \pi_k N(x_n|\mu_k, \Sigma_k) \right) - \lambda \sum_{k=1}^{K} \phi(\pi_k; \Sigma_k)
\]

- EM Algorithm

  - **E-step**: Evaluate the responsibilities (posterior probability of data point \( i \) belonging to mixture component \( k \))
  - **M-step**: Use the updated responsibilities to re-estimate the parameters \( \theta = (\pi_k; \mu_k, \Sigma_k) \)

- Updating mixture weights is as follows if \( \phi \equiv 0 \)

\[
\max_{\pi} \sum_{k=1}^{K} r_k \ln \pi_k \quad \text{subject to} \quad \sum_{k=1}^{K} \pi_k = 1 \]

\( \text{No sparsity is imposed!} \)
Outline

▪ Problem setting

▪ Mixture weight update formulation

▪ Quadratic time algorithm

▪ Experimental results
Mixed-integer Programming for Mixture Weights

- A possible sparse mixture weight updating equation is

\[
\max_{\pi} \sum_{k=1}^{K} a_k \ln(\pi_k)
\]

\[
\text{s.t. } \sum_{k=1}^{K} \pi_k = 1, \quad \pi_k \geq 0
\]

- A convex mixed-integer programming (MIP) reformulation is

\[
\min_{\pi, y} - \sum_{k=1}^{K} a_k \ln(\pi_k) + \tau \sum_{k=1}^{K} y_k
\]

\[
\text{s.t. } \sum_{k=1}^{K} \pi_k = 1, \quad \pi_k \geq 0,
\]

\[
y_k \geq \pi_k, \quad y_k \in \{0, 1\}, \quad k = 1, \ldots, K
\]
Mixed-integer Programming for Mixture Weights

- A possible sparse mixture weight updating equation is

\[
\max_\pi \sum_{k=1}^{K} a_k \ln(\pi_k) - \tau \|\pi\|_0
\]

subject to \( \sum_{k=1}^{K} \pi_k = 1, \pi_k \geq 0 \)

- A convex mixed-integer programming (MIP) reformulation is

\[
\min_{\pi, y} - \sum_{k=1}^{K} a_k \ln(\pi_k) + \tau \sum_{k=1}^{K} y_k
\]

subject to \( \sum_{k=1}^{K} \pi_k = 1, \pi_k \geq 0, \)

\( y_k \geq \pi_k, y_k \in \{0, 1\}, k = 1, \ldots, K \)
Mixed-integer Programming for Mixture Weights

- A possible sparse mixture weight updating equation is

$$\max_{\pi} \sum_{k=1}^{K} a_k \ln(\pi_k) - \tau \|\pi\|_0$$

subject to

$$\sum_{k=1}^{K} \pi_k = 1, \quad \pi_k \geq 0$$

- A convex mixed-integer programming (MIP) reformulation is

$$\min_{\pi, y} -\sum_{k=1}^{K} a_k \ln(\pi_k) + \tau \sum_{k=1}^{K} y_k$$

subject to

$$\sum_{k=1}^{K} \pi_k = 1, \quad \pi_k \geq 0,$$

$$y_k \geq \pi_k, \quad y_k \in \{0, 1\}, \quad k = 1, \ldots, K$$
Definition 1. For a given small $\epsilon > 0$, a vector $x$ is called an $\epsilon$-sparse solution if many elements satisfy $|x_i| \leq \epsilon$.

- A convex mixed-integer programming (MIP) reformulation is

$$\min_{\pi, y} \quad f(\pi, y) \equiv -\sum_{k=1}^{K} a_k \ln(\pi_k) + \tau \sum_{k=1}^{K} y_k$$

s.t. \quad \sum_{k=1}^{K} \pi_k = 1, \pi_k \geq 0,

\quad y_k \geq \pi_k - \epsilon

\quad y_k \in \{0, 1\}, k = 1, \ldots, K.$$
Mixed-integer Programming for Mixture Weights

**Definition 1.** For a given small $\epsilon > 0$, a vector $x$ is called an $\epsilon$-sparse solution if many elements satisfy $|x_i| \leq \epsilon$.

- A convex mixed-integer programming (MIP) reformulation is

$$\min_{\pi, y} \quad f(\pi, y) \equiv -\sum_{k=1}^{K} a_k \ln(\pi_k) + \tau \sum_{k=1}^{K} y_k$$

subject to

$$\sum_{k=1}^{K} \pi_k = 1, \pi_k \geq 0,$$

$$y_k \geq \pi_k - \epsilon,$$

$$y_k \in \{0, 1\}, k = 1, \ldots, K.$$
Outline

▪ Problem setting
▪ Mixture weight update formulation
  ▪ Quadratic time algorithm
▪ Experimental results
Algorithm for the $\mathcal{E}$-sparse Problem

- We assume that

$$0 < a_1 \leq a_2 \leq \ldots \leq a_K,$$

and if $a_i = a_j$ for $i < j$ then $\pi_i \leq \pi_j$

- Denote $\|y\|_\#$ by the number of zero elements of $y$

- We have

(i) If $\|y\|_\# = m$ then $y_1 = \ldots = y_m = 0$ and $y_{m+1} = \ldots = y_K = 1$.
(ii) It holds that $\pi_k \leq \pi_l$ for every $1 \leq k < l \leq K$. 
Algorithm for the $\mathcal{E}$-sparse Problem

- One of hidden parameters for the optimal solution of MIP is the number of zero elements $m = \|y\|\#$. We parameterize it using the parameter $m$.

- When $m$ is given, the MIP is reduced to

$$\begin{align*}
\min_{\pi} & \quad - \sum_{k=1}^{K} a_k \ln(\pi_k) \\
\text{s.t.} & \quad \sum_{k=1}^{K} \pi_k = 1, \\
& \quad \pi_k \leq \epsilon, \quad k = 1, \ldots, m, \\
& \quad \pi_k > \epsilon, \quad k = m + 1, \ldots, K
\end{align*}$$

- We can use exhaustive search for $m = 0, \ldots, K - 1$
Algorithm for the $\epsilon$-sparse Problem

- We propose an alternative, which can be analytically solved for a fixed value $m$

\[
\min_{\pi} \quad - \sum_{k=1}^{K} a_k \ln(\pi_k) \\
\text{s.t.} \quad \sum_{k=1}^{K} \pi_k = 1, \quad \pi_k \leq \epsilon, k = 1, \ldots, m
\]

- Let us define

\[
g(\pi) = - \sum_{k=1}^{K} a_k \ln(\pi_k) + \tau \left| \{i : \pi_i > \epsilon\} \right|
\]

- We need to search for $m$ giving the smallest value for $g(\pi)$
Algorithm for the $\mathcal{E}$-sparse Problem

- The Karush-Kuhn-Tucker (KKT) conditions read

\[
\frac{a_k}{\pi_k} = \begin{cases} 
\lambda, & \text{if } k > m \\
\lambda + \mu_k, & \text{if } k \leq m
\end{cases}
\]

\[
\mu_k(\pi_k - \epsilon) = 0, \quad k \leq m
\]

\[
\mu_k \geq 0, \quad k \leq m.
\]

- **Lemma 1.** The following holds

  (i) If $a_k \geq \epsilon$ and $k \leq m$ then we have $\pi_k = \epsilon$.

  (ii) If $a_m \leq \epsilon$ or $m = 0$ then $\pi = a$.

  (iii) $0 < \pi_1 \leq \pi_2 \leq \ldots \leq \pi_K$
Algorithm for the $\mathcal{E}$-sparse Problem

- For a given $m$, we need to identify a break-point $\hat{k}$ where

$$\pi_k < \epsilon, \text{ if } k \leq \hat{k}$$

$$\pi_k = \epsilon, \text{ if } \hat{k} < k \leq m$$

(1)

- For any $k \leq \hat{k}$ or $k > m$, one has

$$\pi_k = \frac{a_k (1 - (m - \hat{k}) \epsilon)}{\sum_{i \leq \hat{k} \text{ or } i > m} a_i}$$

(2)
Algorithm for the $\mathcal{E}$-sparse Problem

**Algorithm 1** Sparse Weight Selection Algorithm - $\text{SWSA}(a, \tau, \epsilon)$

```plaintext
Set $f_{\text{min}} \leftarrow -\sum_{k=1}^{K} a_k \ln(a_k) + n\tau$
for $m = 0, 1, \ldots, n - 1$ do
  if $m = 0$ or $a_m \leq \epsilon$ then
    $\pi \leftarrow a$
  else
    Find $t \leq m$ such that $a_t < \epsilon \leq a_{t+1}$
    if $t = 0$ then
      $\pi_k \leftarrow \left\{ \begin{array}{ll}
      \epsilon, & \text{if } k \leq m \\
      \frac{a_k(1-m\epsilon)}{\sum_{i=m+1}^{n} a_i}, & \text{otherwise}
      \end{array} \right.$
    else
      for $\hat{k} = t, t-1, \ldots, 1$ do
        $\pi_k \leftarrow$ Eqs. (1) and (2)
        if $\left( \pi_k < \epsilon \text{ and } a_{k+1} (1 - (m - \hat{k})\epsilon) \geq \epsilon \sum_{i \leq \hat{k} \text{ or } i > m} a_i \right)$ then
          break
        end if
      end for
    end if
  end if
Compute $g(\pi) \leftarrow -\sum_{k=1}^{K} a_k \ln(\pi_k) + \tau |\{i : \pi_i > \epsilon\}|$
if $g(\pi) < f_{\text{min}}$ then
  $f_{\text{min}} \leftarrow g(\pi)$ and $\pi^* \leftarrow \pi$
end if
end if
end for
return $\pi^*$
```

**Theorem.** Algorithm 1 can find a global optimal solution of the MIP in quadratic time in terms of the maximum number of mixture components $K$. 
Outline

- Problem setting
- Mixture weight update formulation
- Quadratic time algorithm
- Experimental results
Experiment (1): Learning variable-variable dependency from data (synthetic data)

Ground truth

Proposed method: able to recover the ground truth

Our method successfully reproduced the ground truth

Conventional method: inaccurate
Experiment (1): Learning variable-variable dependency from data (synthetic data)

Ground truth

Our method successfully reproduced the ground truth

Proposed method: able to recover the ground truth

Conventional method: inaccurate
## Experiment (2): Performance comparison for held-out log probability

<table>
<thead>
<tr>
<th></th>
<th>log-like.</th>
<th>BICscore</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Breast Cancer</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SWSA</td>
<td>276.75</td>
<td>1153.72</td>
<td>5.3</td>
</tr>
<tr>
<td>CARD</td>
<td>230.94</td>
<td>907.45</td>
<td>14.2</td>
</tr>
<tr>
<td>CSBDP</td>
<td>203.51</td>
<td>-</td>
<td>12.6</td>
</tr>
<tr>
<td>VDP</td>
<td>224.86</td>
<td>-</td>
<td>7.5</td>
</tr>
<tr>
<td><strong>Cloud</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SWSA</td>
<td>262.88</td>
<td>2410.21</td>
<td>7.4</td>
</tr>
<tr>
<td>CARD</td>
<td>228.11</td>
<td>1905.44</td>
<td>10.1</td>
</tr>
<tr>
<td>CSBDP</td>
<td>249.51</td>
<td>-</td>
<td>7.8</td>
</tr>
<tr>
<td>VDP</td>
<td>231.02</td>
<td>-</td>
<td>6.6</td>
</tr>
<tr>
<td><strong>Parkinsons</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SWSA</td>
<td>-89.02</td>
<td>-3615.72</td>
<td>2.6</td>
</tr>
<tr>
<td>CARD</td>
<td>-107.53</td>
<td>-5135.02</td>
<td>5.8</td>
</tr>
<tr>
<td>CSBDP</td>
<td>-98.61</td>
<td>-</td>
<td>6.5</td>
</tr>
<tr>
<td>VDP</td>
<td><strong>-86.09</strong></td>
<td>-</td>
<td><strong>2.4</strong></td>
</tr>
<tr>
<td><strong>Anuran Calls</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SWSA</td>
<td>2592.58</td>
<td>42528.4</td>
<td>3.2</td>
</tr>
<tr>
<td>CARD</td>
<td>2357.89</td>
<td>37413.6</td>
<td>11.6</td>
</tr>
<tr>
<td>CSBDP</td>
<td>2426.55</td>
<td>-</td>
<td>13.7</td>
</tr>
<tr>
<td>VDP</td>
<td>2386.32</td>
<td>-</td>
<td>3.8</td>
</tr>
</tbody>
</table>

SWSA : Proposed method  
CARD : Conventional method  
VDP : Variational Dirichlet process [Blei and Jordan, 2006]  
CSBDP : Collapsed variational stick-breaking Dirichlet process [Kurihara et al. 2007]
Experiment (3): Anomaly Detection

Multi-variate data

Sparse graphical model

Anomaly score

\[ a(x) = -\ln p(x \mid D) \]

<table>
<thead>
<tr>
<th></th>
<th>SWSA</th>
<th>CARD</th>
<th>T2</th>
<th>PCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wafer</td>
<td>0.96</td>
<td>0.88</td>
<td>0.86</td>
<td>0.81</td>
</tr>
<tr>
<td>Letter</td>
<td>0.97</td>
<td>0.91</td>
<td>0.85</td>
<td>0.84</td>
</tr>
</tbody>
</table>
Experiment (4): Scalability Comparison

The graph shows the running time (in seconds) for different methods as a function of the number of mixing weights $K$. The methods compared are:

- **proposed**
- NLP
- MIP

The x-axis represents the number of mixing weights $K$ ranging from $10^1$ to $10^4$, while the y-axis represents the running time from $10^{-4}$ to $10^3$.
Conclusions

- Introduced a new formulation for updating sparse mixture weights
- Developed a quadratic time algorithm
- Demonstrated the good performance for both synthetic and real datasets
THANK YOU!