IBM **Research**

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

Gaussian Mixture Model

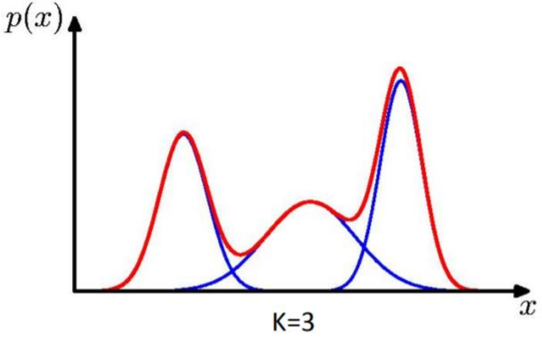
- A Gaussian mixture model (GMM) is a weighted sum of *K* component Gaussian densities

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Component

Mixing coefficient

$$\forall k : \pi_k \geqslant 0 \qquad \sum_{k=1}^{K} \pi_k = 1$$



  - π_k : mixture weights
  - (μ_k, Σ_k) : the mean and covariance

- Irrelevant components can be mistakenly included in the training model

Sparse Mixture of Sparse Gaussian Graphical Model

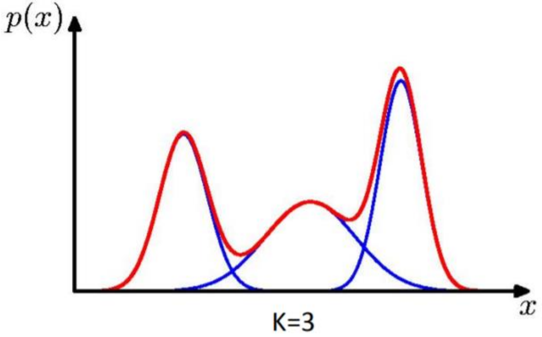
- A Gaussian mixture model (GMM) is a weighted sum of *K* component Gaussian densities

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Component: $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

Mixing coefficient: π_k

$$\forall k : \pi_k \geqslant 0 \qquad \sum_{k=1}^{K} \pi_k = 1$$



  - π_k : sparse mixture weights
  - Σ_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

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

- 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 $\phi \equiv 0$

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

Expectation-Maximization (EM) Algorithm

- We suggest a penalized log-likelihood as

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

- 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 $\phi \equiv 0$

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

⇦ 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

$$\max_{\boldsymbol{\pi}} \sum_{k=1}^{K} a_k \ln(\pi_k)$$
$$\text{s.t.} \quad \textstyle\sum_{k=1}^{K} \pi_k = 1, \quad \pi_k \geq 0$$

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

$$\min_{\boldsymbol{\pi},\boldsymbol{y}} \quad -\textstyle\sum_{k=1}^{K} a_k \ln(\pi_k) + \tau \textstyle\sum_{k=1}^{K} y_k$$
$$\text{s.t.} \quad \textstyle\sum_{k=1}^{K} \pi_k = 1, \pi_k \geq 0,$$
$$y_k \geq \pi_k, y_k \in \{0, 1\}, k = 1, \ldots, K$$

Mixed-integer Programming for Mixture Weights

- A possible sparse mixture weight updating equation is

$$\max_{\boldsymbol{\pi}} \sum_{k=1}^{K} a_k \ln(\pi_k) - \tau \|\boldsymbol{\pi}\|_0$$
$$\text{s.t.} \quad \textstyle\sum_{k=1}^{K} \pi_k = 1, \quad \pi_k \geq 0$$

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

$$\min_{\boldsymbol{\pi},\boldsymbol{y}} \quad -\textstyle\sum_{k=1}^{K} a_k \ln(\pi_k) + \tau \textstyle\sum_{k=1}^{K} y_k$$
$$\text{s.t.} \quad \textstyle\sum_{k=1}^{K} \pi_k = 1, \pi_k \geq 0,$$
$$y_k \geq \pi_k, y_k \in \{0, 1\}, k = 1, \ldots, K$$

Mixed-integer Programming for Mixture Weights

- A possible sparse mixture weight updating equation is

$$\max_{\boldsymbol{\pi}} \sum_{k=1}^{K} a_k \ln(\pi_k) - \tau \|\boldsymbol{\pi}\|_0$$
$$\text{s.t.} \quad \sum_{k=1}^{K} \pi_k = 1, \quad \pi_k \geq 0$$

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

$$\min_{\boldsymbol{\pi}, \boldsymbol{y}} \quad -\sum_{k=1}^{K} a_k \ln(\pi_k) + \tau \sum_{k=1}^{K} y_k$$
$$\text{s.t.} \quad \sum_{k=1}^{K} \pi_k = 1, \pi_k \geq 0,$$
$$y_k \geq \pi_k, y_k \in \{0, 1\}, k = 1, \ldots, K$$

Mixed-integer Programming for Mixture Weights

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

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

$$\begin{aligned} \min_{\boldsymbol{\pi},\boldsymbol{y}} \quad & f(\boldsymbol{\pi},\boldsymbol{y}) \equiv -\textstyle\sum_{k=1}^{K} a_k \ln(\pi_k) + \tau \sum_{k=1}^{K} y_k \\ \text{s.t.} \quad & \textstyle\sum_{k=1}^{K} \pi_k = 1, \pi_k \geq 0, \\ & y_k \geq \pi_k - \epsilon \\ & y_k \in \{0,1\}, k = 1,\ldots,K. \end{aligned}$$

Mixed-integer Programming for Mixture Weights

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

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

$$\begin{aligned}
\min_{\boldsymbol{\pi}, \boldsymbol{y}} \quad & f(\boldsymbol{\pi}, \boldsymbol{y}) \equiv -\sum_{k=1}^{K} a_k \ln(\pi_k) + \tau \sum_{k=1}^{K} y_k \\
\text{s.t.} \quad & \sum_{k=1}^{K} \pi_k = 1, \pi_k \geq 0, \\
& y_k \geq \pi_k - \epsilon \\
& y_k \in \{0, 1\}, k = 1, \ldots, K.
\end{aligned}$$

Outline

- Problem setting
- Mixture weight update formulation
- Quadratic time algorithm
- Experimental results

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

- We assume that

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

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

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

- We have

(i) *If* $\|\boldsymbol{y}\|_\# = m$ *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

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

- We can use exhaustive search for $m = 0, \ldots K - 1$

Algorithm for the ε-sparse Problem

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

$$\begin{aligned} \min_{\boldsymbol{\pi}} \quad & -\sum_{k=1}^{K} a_k \ln(\pi_k) \\ \text{s.t.} \quad & \sum_{k=1}^{K} \pi_k = 1, \\ & \pi_k \le \epsilon, k = 1, \ldots, m \end{aligned}$$

- Let us define

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

- We need to search for *m* giving the smallest value for $g(\boldsymbol{\pi})$

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

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

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

$$\mu_k(\pi_k - \epsilon) = 0, \quad k \le m$$

$$\mu_k \ge 0, \quad k \le m.$$

- ***Lemma 1.*** The following holds

(i) *If* $a_k \ge \epsilon$ *and* $k \le m$ *then we have* $\pi_k = \epsilon$.

(ii) *If* $a_m \le \epsilon$ *or* $m = 0$ *then* $\boldsymbol{\pi} = \boldsymbol{a}$.

(iii) $0 < \pi_1 \le \pi_2 \le \ldots \le \pi_K$

Algorithm for the ε-sparse Problem

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

$$\pi_k < \epsilon, \text{ if } k \leq \hat{k}$$
$$\pi_k = \epsilon, \text{ if } \hat{k} < k \leq m \qquad (1)$$

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

$$\pi_k = \frac{a_k(1-(m-\hat{k})\epsilon)}{\sum_{i \leq \hat{k} \text{ or } i > m} a_i} \qquad (2)$$

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

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

```
Set f_min ← −Σ_{k=1}^{K} a_k ln(a_k) + nτ
for m = 0, 1, ..., n − 1 do
  if m = 0 or a_m ≤ ε then
    π ← a
  else
    Find t ≤ m such that a_t < ε ≤ a_{t+1}
    if t = ∅ then
      π_k ← { ε,                                   if k ≤ m
            { a_k(1−mε) / Σ_{i=m+1}^{n} a_i,       otherwise
    else
      for k̂ = t, t − 1, ..., 1 do
        π_k ← Eqs. (1) and (2)
        if ( π_k̂ < ε and a_{k̂+1}(1 − (m − k̂)ε) ≥ ε Σ_{i≤k̂ or i>m} a_i ) then
          break
        end if
      end for
    end if
    Compute g(π) ← −Σ_{k=1}^{K} a_k ln(π_k) + τ|{i : π_i > ε}|
    if g(π) < f_min then
      f_min ← g(π) and π* ← π
    end if
  end if
end for
return π*
```

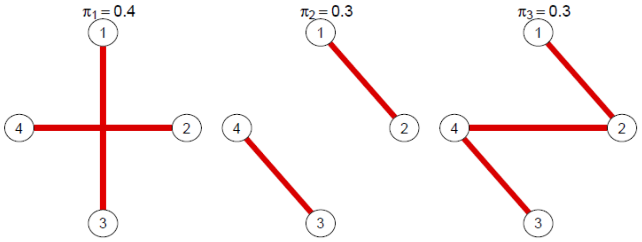
Theorem. Algorithm 1 can find a global optimal solution of the MIP in quadratic time in terms of the maximum number of mixture components K.

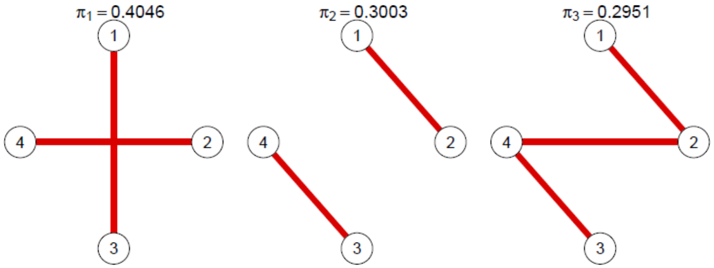
Outline

- Problem setting
- Mixture weight update formulation
- Quadratic time algorithm
- Experimental results

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

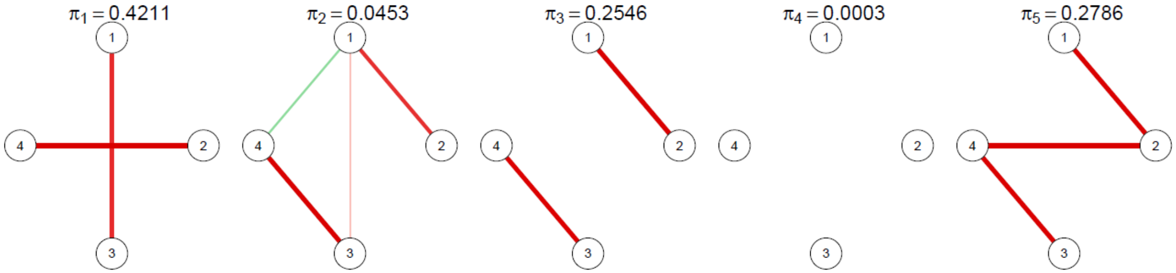
Ground truth





Proposed method: able to recover the ground truth

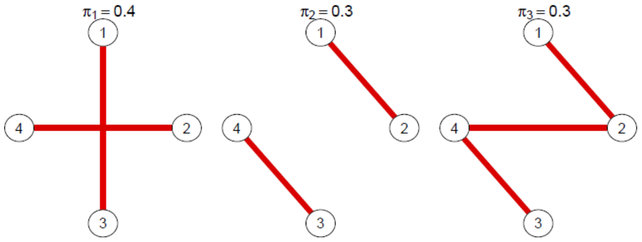
Our method successfully reproduced the ground truth

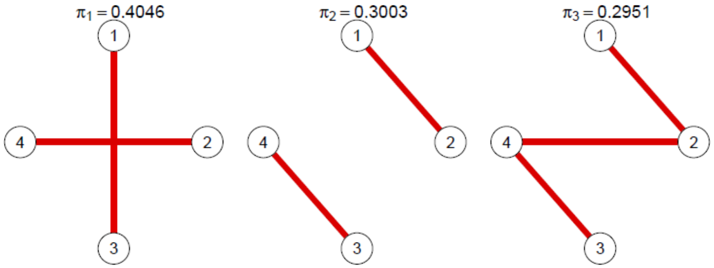


Conventional method: inaccurate

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

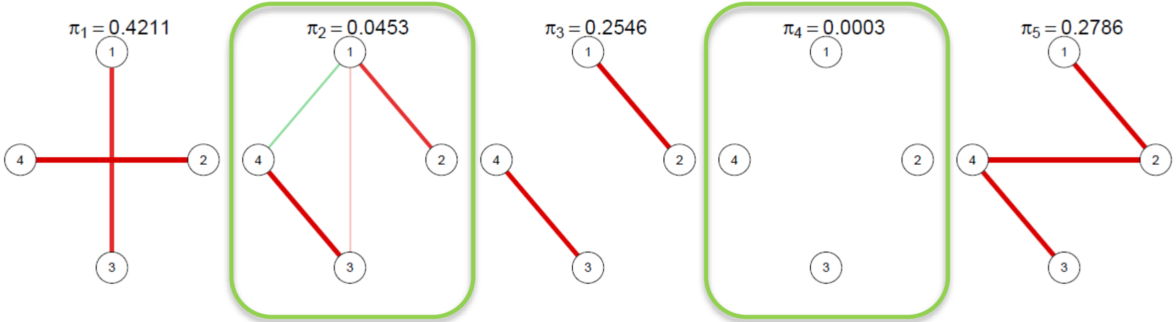
Ground truth





Proposed method: able to recover the ground truth

Our method successfully reproduced the ground truth



Conventional method: inaccurate

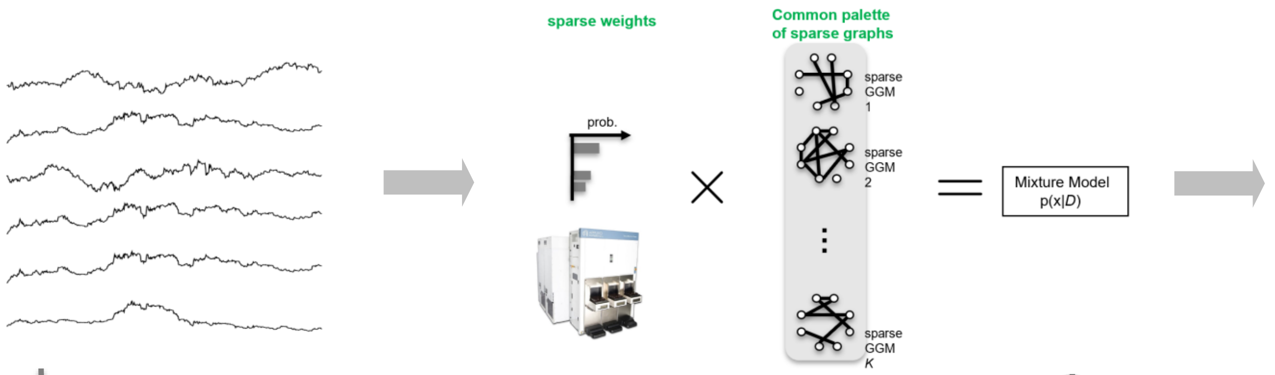
Experiment (2): Performance comparison for held-out log probability

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

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**



$$a(\boldsymbol{x}) = -\ln p(\boldsymbol{x} \mid \mathcal{D})$$

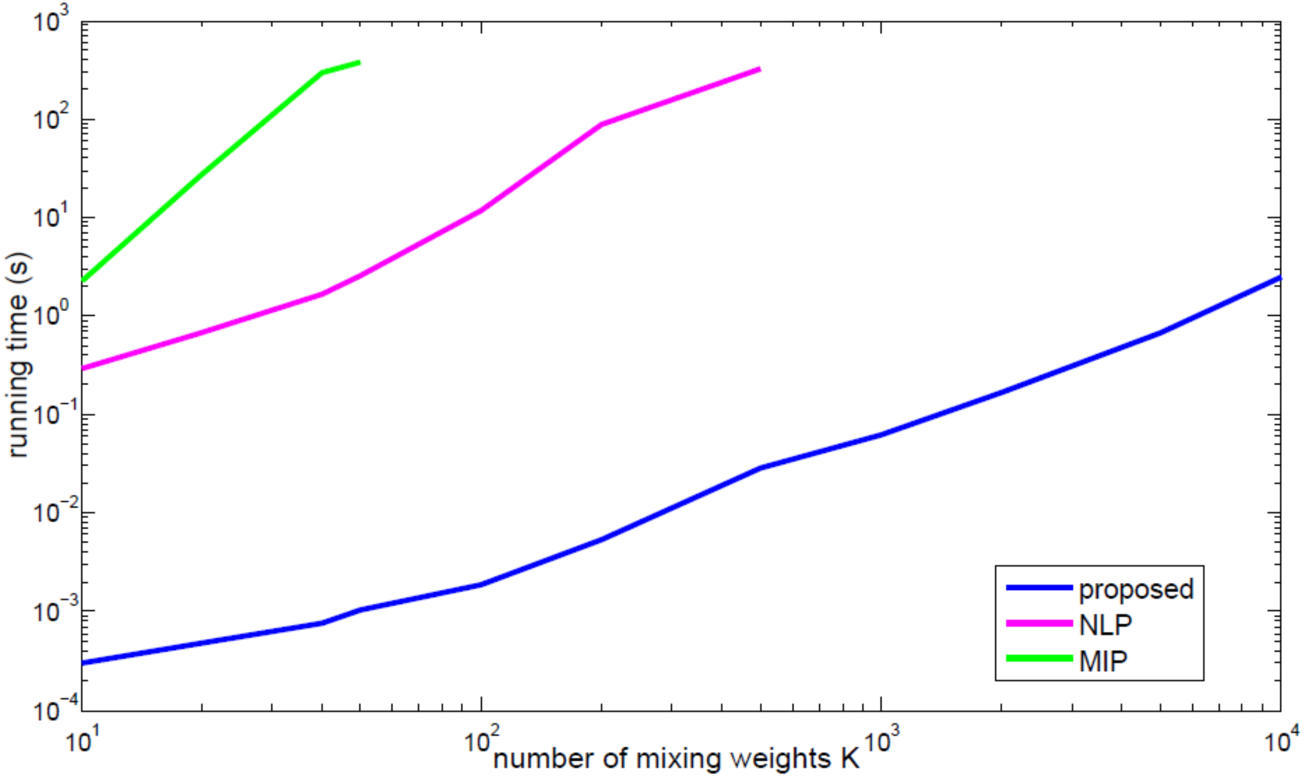
training data

testing data

	SWSA	CARD	T2	PCA
Wafer	**0.96**	0.88	0.86	0.81
Letter	**0.97**	0.91	0.85	0.84

Comparison of AUC performance

Experiment (4): Scalability Comparison



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!