
Multivariate Density Estimation with Permuted Variable-Values

Sridevi Parise
Padhraic Smyth
Sergey Kirshner

*School of Information and Computer Science
University of California, Irvine*

Outline

- Permuted data
- Learning problems with permuted data
- Characterization of the difficulty
- Learning algorithms
- Conclusions

Permuted Data

f_1	f_2	f_3	Permuted Data			Permutation
23	72	150	150	72	23	(3, 2, 1)
25	64	107	25	107	64	(1, 3, 2)
52	68	200	200	68	52	(3, 2, 1)
37	84	270	84	270	37	(2, 3, 1)

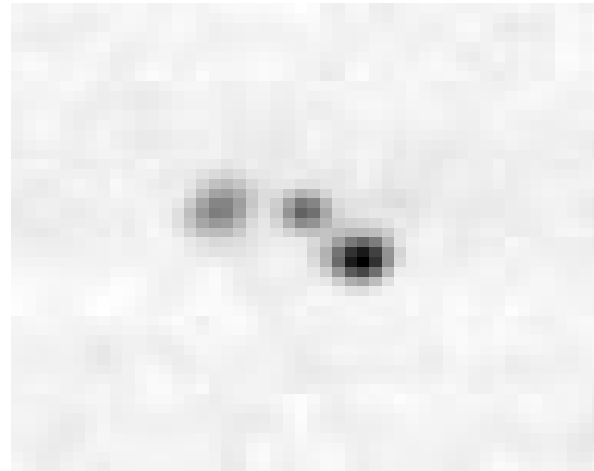
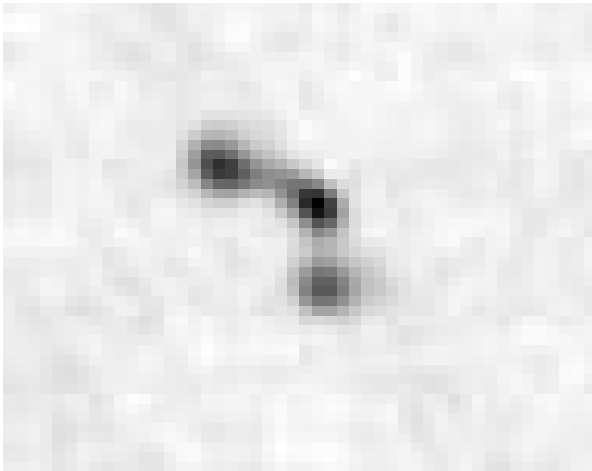
Problems of Interest

Given permuted data and the set of possible permutations:

- How do you *Unscramble* the data?
 - How hard is this problem?
 - Theoretical characterization of this difficulty
- How do you learn the joint density of the features?
 - Algorithms to solve this problem

Motivation

Recent work on astronomical image data (**Kirshner et al., 2003**)



Generative Model

- Generate N data vectors
 - Each data vector \mathbf{x} generated according to $p(\mathbf{x})$, the joint feature density
- For each \mathbf{x} , choose a permutation ρ from $\mathcal{P} = \{\rho_1, \dots, \rho_m\}$ according to $p(\rho)$
- Permute \mathbf{x} according to ρ

Probability Densities for Permuted Data

Given a permuted vector \mathbf{x} ,

$$\tilde{p}(\mathbf{x}) = \sum_{j=1}^m p(\rho_j) \cdot p(\rho_j^{-1}(\mathbf{x}))$$

Unscrambling the Data

- Assume we know $p(\mathbf{x})$, \mathcal{P} and $p(\rho)$,
- Given a permuted vector \mathbf{x} , identify the permutation that generated \mathbf{x}

Bayes Optimal Decision Rule:

Compute

$$\begin{aligned}\tilde{p}(\rho_j|\mathbf{x}) &\propto \tilde{p}(\mathbf{x}|\rho_j)p(\rho_j) \\ &\propto p(\rho_j^{-1}(\mathbf{x}))p(\rho_j), 1 \leq j \leq m\end{aligned}$$

find $\arg \max_j \tilde{p}(\rho_j|\mathbf{x})$

Optimal Permutation Error Rate, E_P^*

Under this optimal decision rule,
Average error rate,

$$E_P^* = \int_{S^c} \tilde{p}(\mathbf{x}) \cdot \left(1 - \max_j \tilde{p}(\rho_j | \mathbf{x}) \right) d\mathbf{x}$$

We call this the **Bayes Optimal Permutation Error Rate**.

- characterizes the *difficulty* of unscrambling
- Defines the minimum error rate any unscrambling algorithm can achieve
- Intuitively assume E_P^* to depend on feature overlap or similarity

Optimal Feature Classification Error Rate, E_C^*

- Error rate for classifying a feature value x as associated with one of the c features
 - Characterizes column overlap
- Define marginal distribution for each feature i ,

$$p(x|C_i) = \int \dots \int p(x_1, \dots, x_{i-1}, x, x_{i+1}, \dots, x_c) dx_1 \dots dx_{i-1} dx_{i+1} \dots dx_c,$$

Therefore,

$$p(x) = \sum_{i=1}^c p(x|C_i) \cdot p(C_i)$$

where $p(C_i) = \frac{1}{c}, 1 \leq i \leq c$

Optimal Feature Classification Error Rate (cont.)

Bayes optimal error rate,

$$E_C^* = \int p(x) \cdot \left(1 - \max_j p(C_j|x)\right) dx.$$

We have proved that, under certain assumptions,

$$\mathbf{E}_P^* \leq \mathbf{E}_C^*$$

$$E_P^* \leq E_C^*$$

Definition: For \mathcal{P} , let k be a **key** index if $(\rho_1(k), \dots, \rho_m(k))$ is a permutation of $(1, \dots, c)$.

Example: $\mathcal{P} = \{\rho_1, \rho_2, \rho_3, \rho_4\}$, $k = 2$

ρ_1	1	2	3	4
ρ_2	2	1	3	4
ρ_3	1	3	4	2
ρ_4	2	4	1	3

Theorem: Given a set of permutations \mathcal{P} with a key and $p(\rho_i) = \frac{1}{c}$, then,

$$E_P^* \leq E_C^*$$

E_P^* for some special cases

Case 1: $c = 2$, $S = \mathbb{R}$ and $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\bar{\mu}, \Sigma)$

where,

$$\bar{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \text{ and } \Sigma = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}$$

$\rho_1 = (1, 2)$ and $\rho_2 = (2, 1)$,

$$\tilde{p}(\mathbf{x}|\rho_1) = p(x_1, x_2) = \mathcal{N}((x_1, x_2)|\bar{\mu}, \Sigma)$$

$$\tilde{p}(\mathbf{x}|\rho_2) = p(x_2, x_1) = \mathcal{N}((x_1, x_2)|\bar{\mu}', \Sigma)$$

where

$$\bar{\mu}' = \begin{bmatrix} \mu_2 \\ \mu_1 \end{bmatrix}$$

It can be shown that (Duda et al.),

$$E_P^* = \frac{1}{\sqrt{2\pi}} \int_{\frac{|\mu_1 - \mu_2|}{\sqrt{2}\sigma}}^{\infty} e^{-u^2/2} du$$

As $|\mu_1 - \mu_2|$ decreases, E_P^* increases and
as σ increases, E_P^* increases

E_P^* for some special cases (Cont.)

Case 2: $c = 2$, $S = \mathbb{R}$ and $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\bar{\mu}, \Sigma)$

where,

$$\bar{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \text{ and } \Sigma = \begin{pmatrix} \sigma^2 & \nu \\ \nu & \sigma^2 \end{pmatrix}, -\sigma^2 < \nu < \sigma^2$$

Proceeding as in case 1,

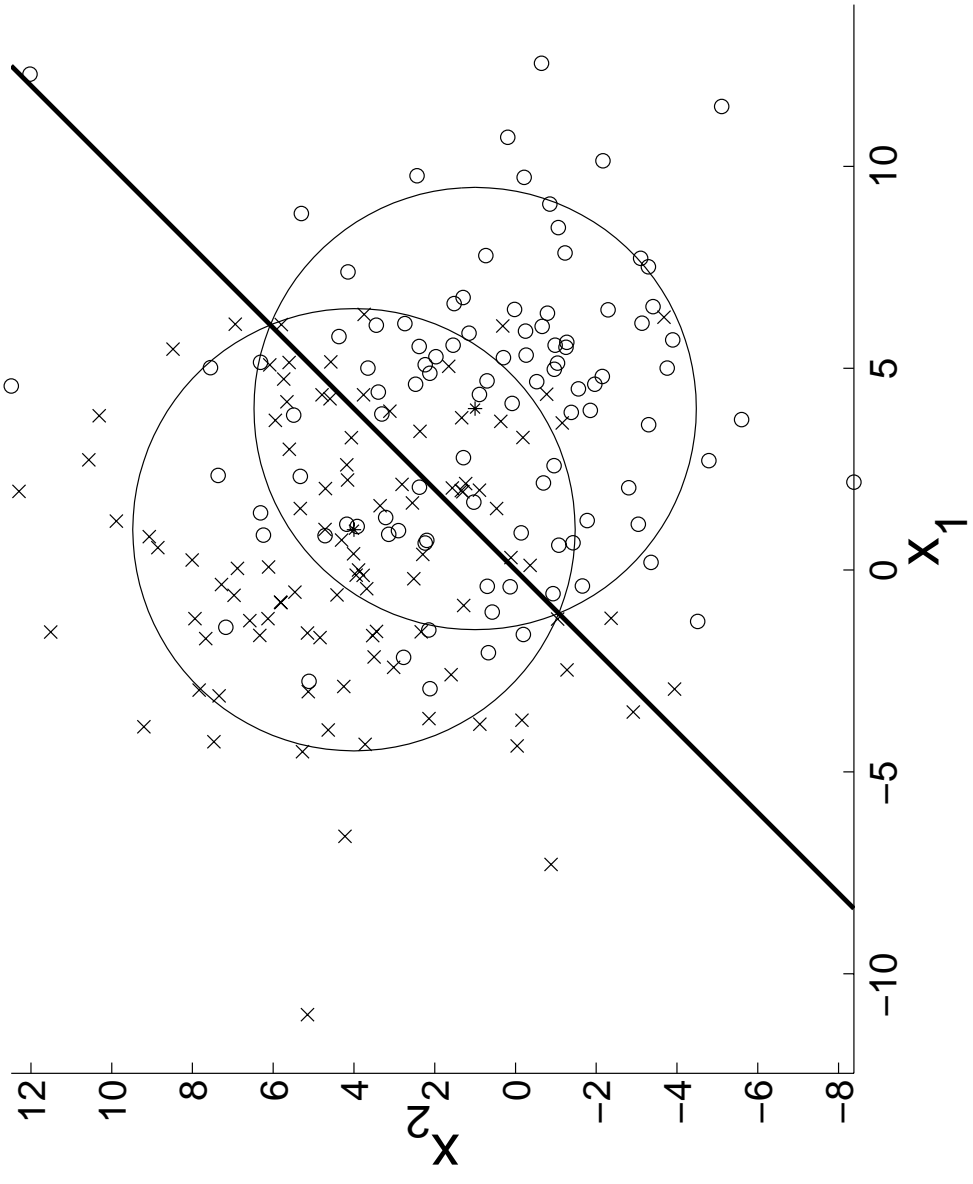
$$E_P^* = \frac{1}{\sqrt{2\pi}} \int_{\frac{|\mu_1 - \mu_2|}{\sqrt{2}\sqrt{\sigma^2 - \nu}}}^{\infty} e^{-u^2/2} du,$$

Therefore as $\nu \rightarrow \sigma^2$, we have $E_P^* \rightarrow 0$

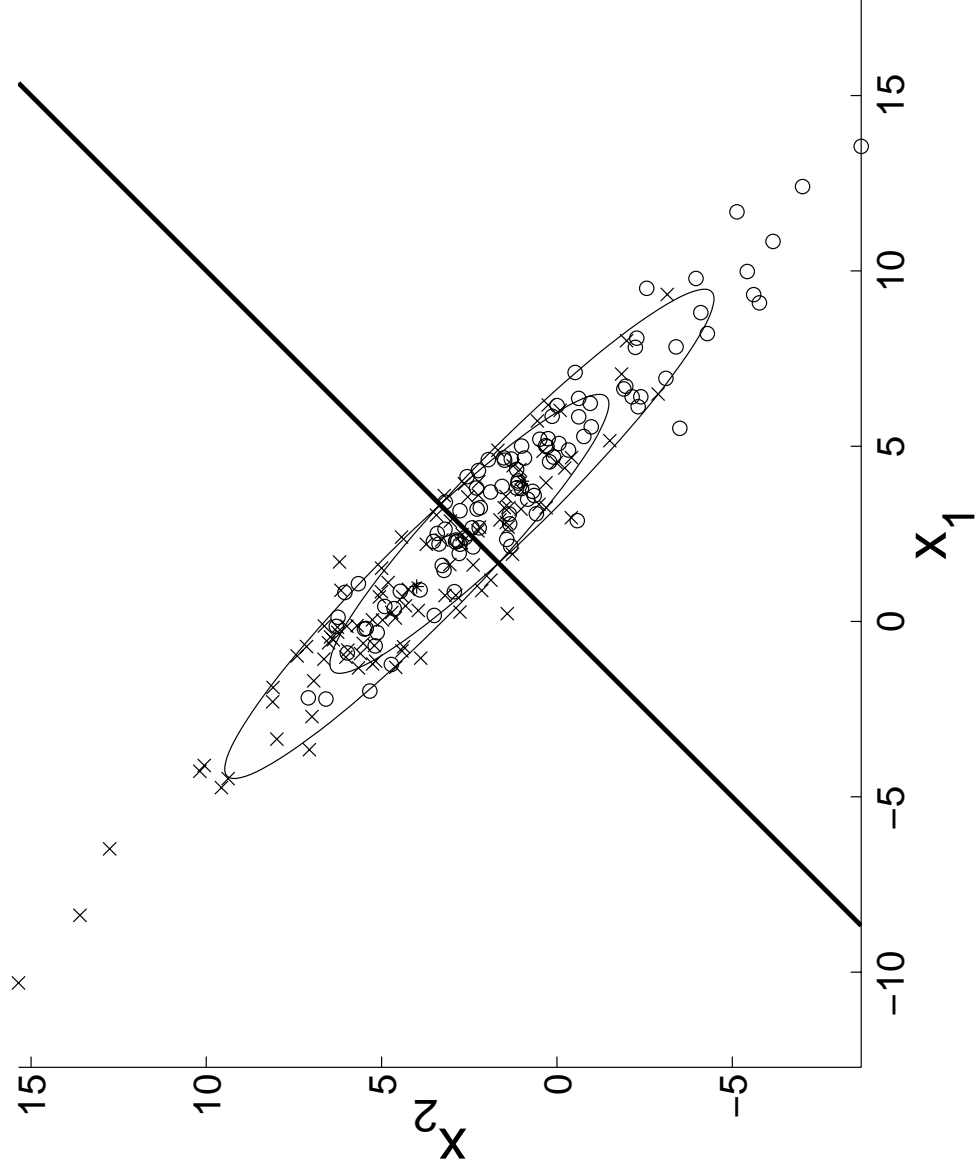
E_P^* for some special cases (Cont.)

From this we can see that,

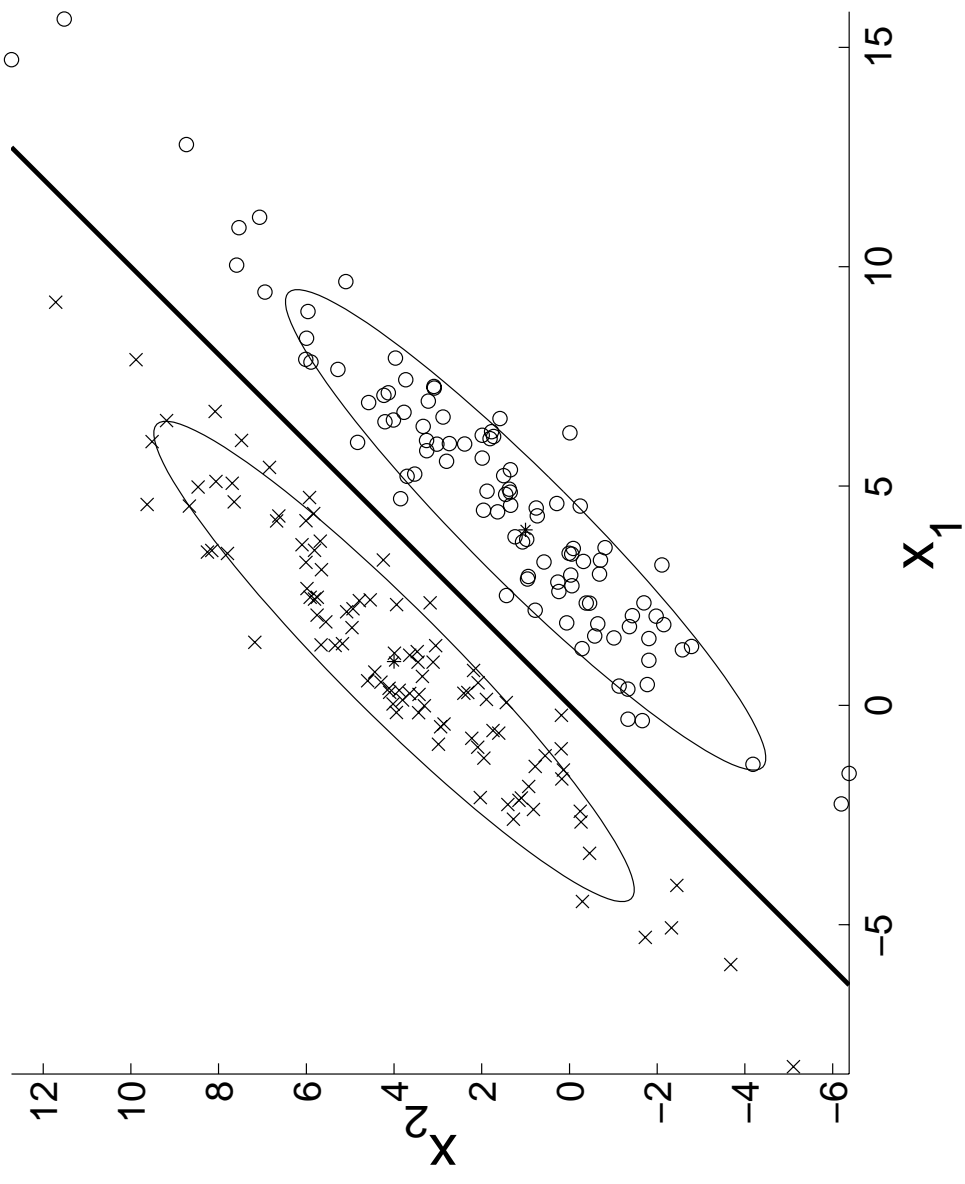
- Negative correlation makes the problem harder
- Even with lot of overlap, positive correlation can make unscrambling easy



$$\nu = 0$$



$$\nu \sim -\sigma^2$$



$$\nu \sim \sigma^2$$

Learning from permuted data using EM

Given \mathcal{P} and the functional form of $p(\mathbf{x})$,

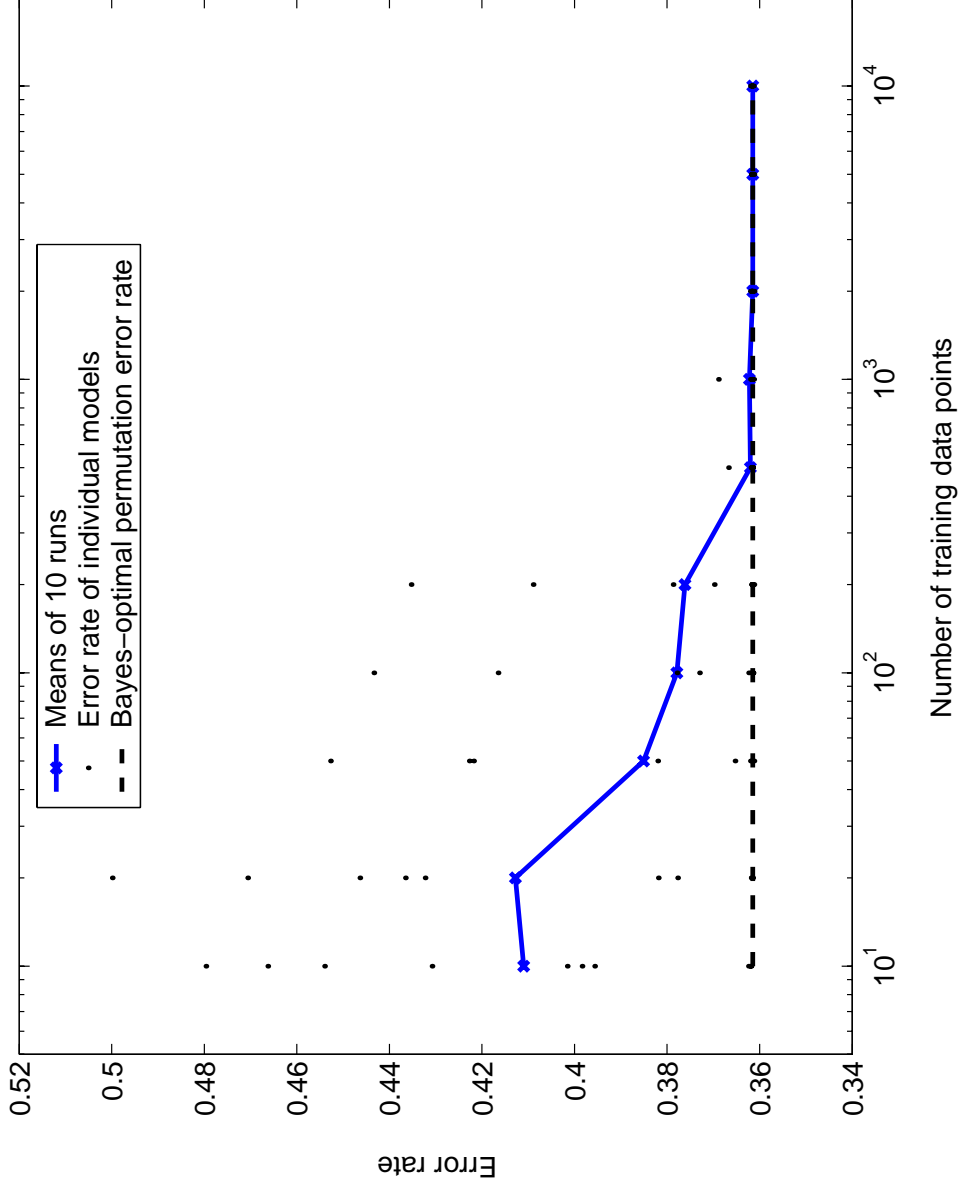
- Estimate the parameters of $p(\mathbf{x})$
- Find the probability that a given permutation ρ has been applied to permuted vector \mathbf{x}

Can be treated as a standard mixture problem and can use EM

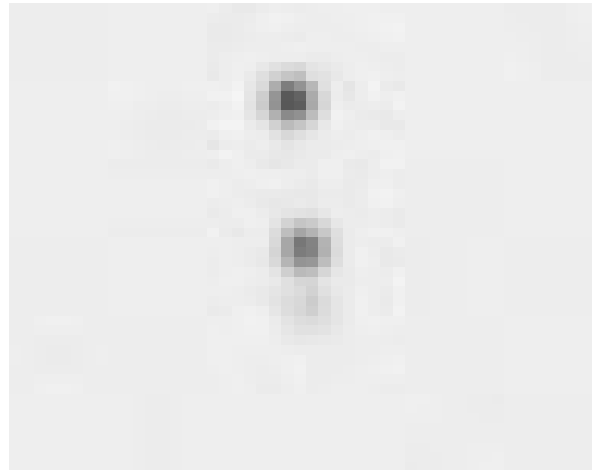
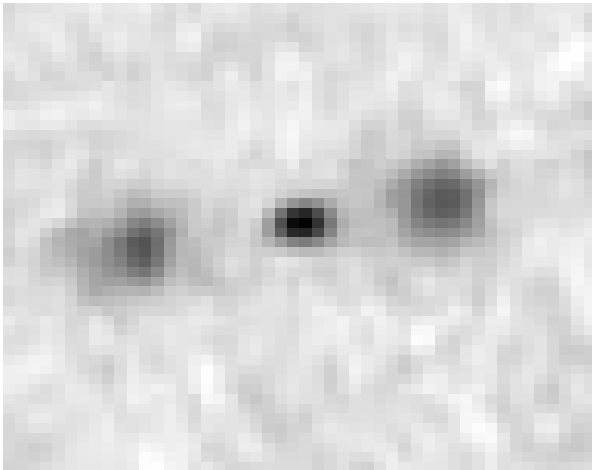
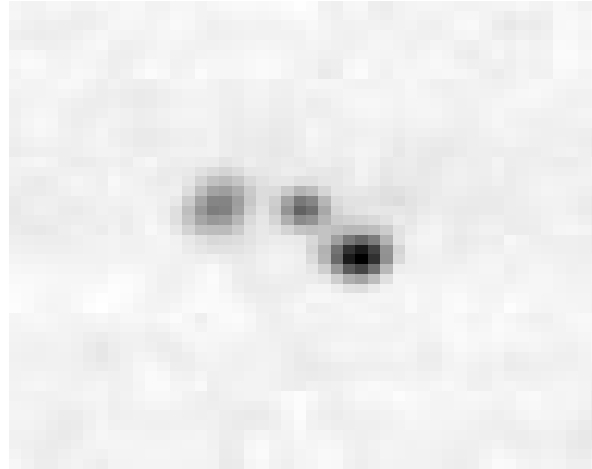
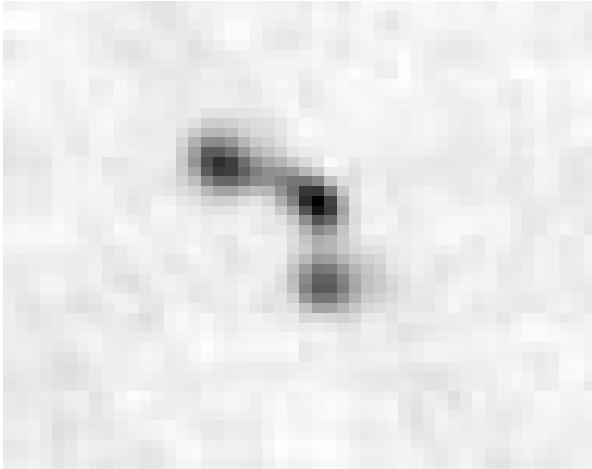
Specifically, if θ denotes the unknown parameters of $p(\mathbf{x})$ and the $p(\rho_j)$'s then,

$$Q(\theta, \theta^{old}) = \sum_{j=1}^m \sum_{i=1}^N \log(p(\rho_j | \theta)) \cdot \tilde{p}(\rho_j | \mathbf{x}^{(i)}, \theta^{old}) \\ + \sum_{j=1}^m \sum_{i=1}^N \log(\tilde{p}(\mathbf{x}^{(i)} | \rho_j, \theta)) \cdot \tilde{p}(\rho_j | \mathbf{x}^{(i)}, \theta^{old})$$

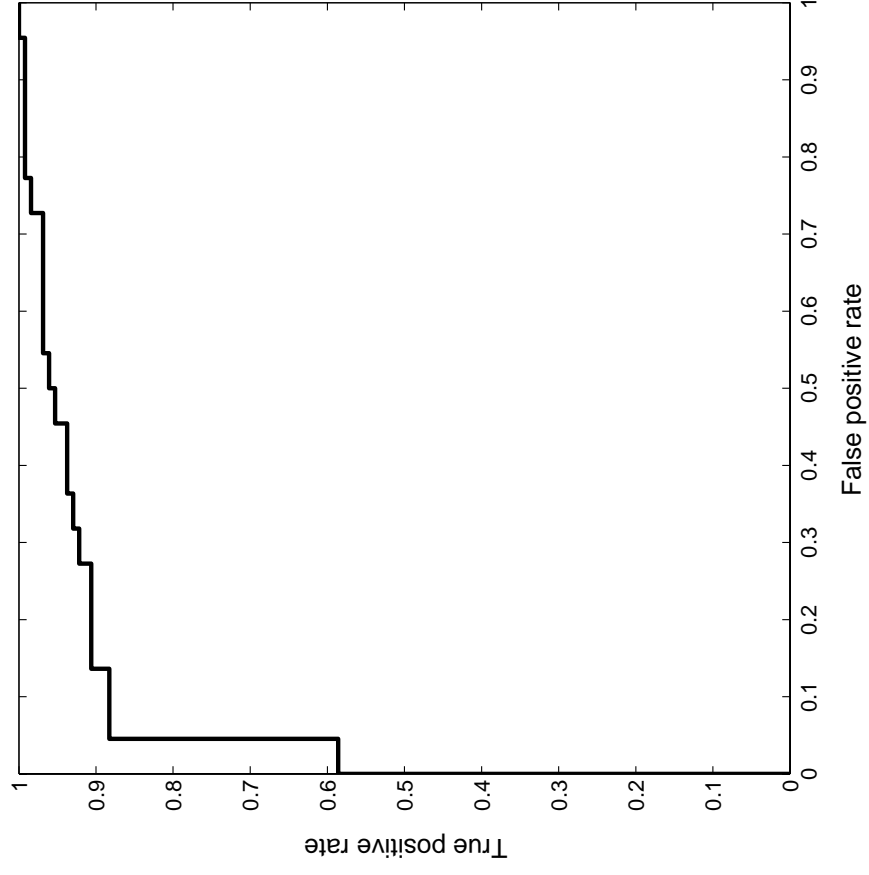
Learning from permuted data (Results)



Results (cont.)



Results (cont.)



Conclusions and Future Work

- Analyzed unsupervised learning with permuted data
- Introduced Bayes Optimal Permutation Error rate, E_P^*
- Closed form expressions for E_P^* in special cases
- Empirical illustration of learning via EM

- Further analysis of the relation between E_P^* and E_C^*
- Learning algorithms for more general transformations

Further information can be found at,

www.datalab.uci.edu