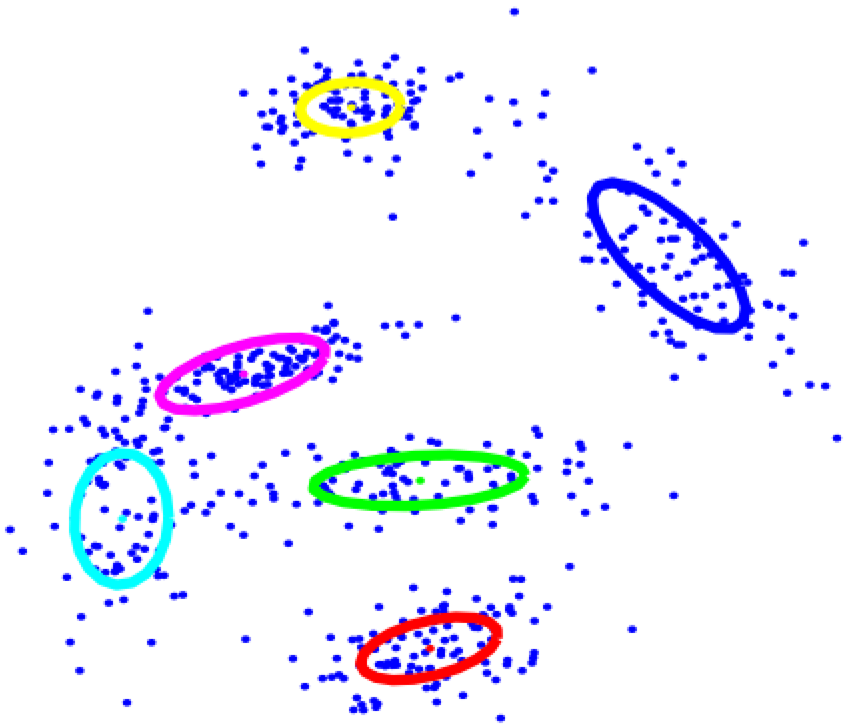
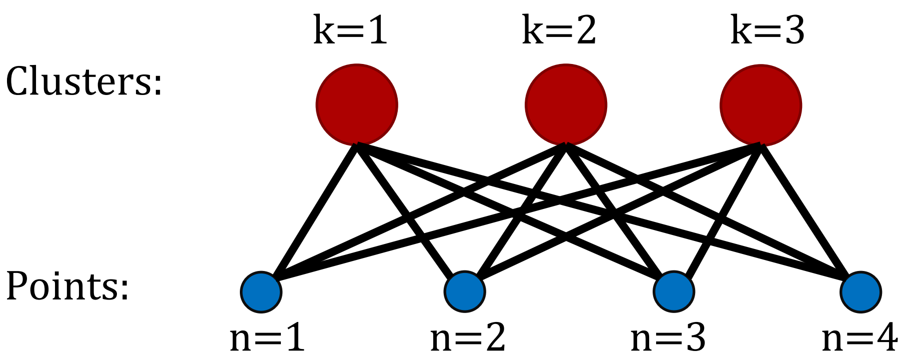
Memoized Online Variational Inference for Dirichlet Process Mixture Models

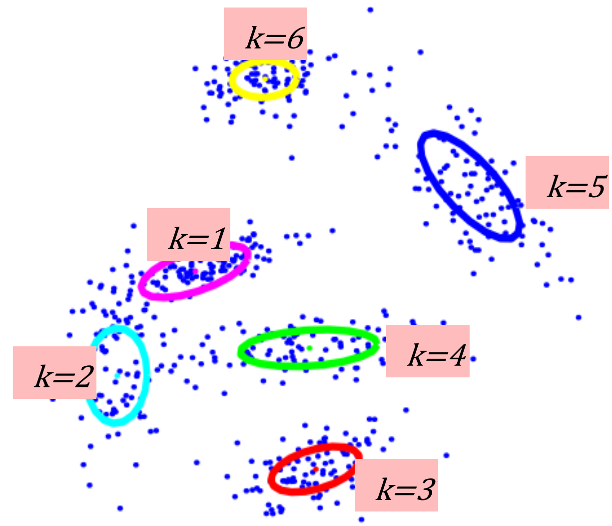
NIPS 2013

Michael C. Hughes and Erik B. Sudderth



Motivation



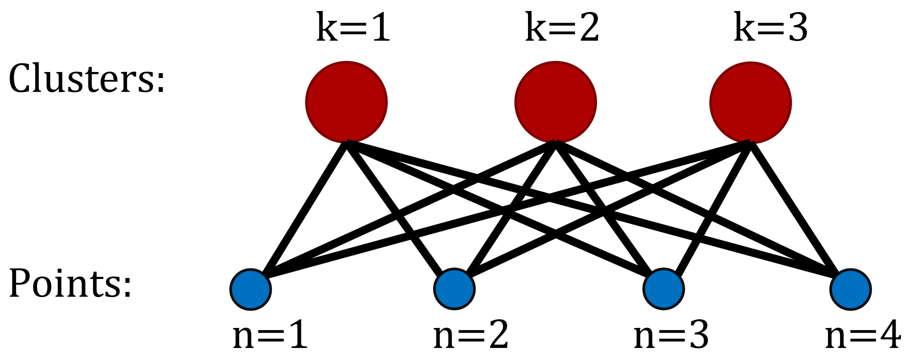


Cluster-point assignment:

$$p(z_n = k)$$

Cluster parameters:

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_K\}$$



Cluster-point assignment:

$$p(z_n = k)$$

Cluster component parameters:

$$\Theta = \{\theta_1, \theta_2, \dots, \theta_K\}$$

$N \gg K$

The usual scenario:

Loop until convergence:

$\boldsymbol{K \times N}$

For $n = 1, \dots, N$, and $k = 1, \dots, K$

$$p(z_n = k) \leftarrow f(\Theta, \mathrm{k}, \mathrm{n})$$

Clustering Assignment Estimation

$\boldsymbol{K}$

For $k = 1, \dots, K$

$$\theta_k \leftarrow g(p(\mathbf{z}),\ k)$$

Component Parameter Estimation

$K \times N$

K

Loop until convergence:

For $n = 1, \ldots, N$, and $k = 1, \ldots, K$
$p(z_n = k) \leftarrow f(\Theta, \mathrm{k}, \mathrm{n})$

For $k = 1, \ldots, K$
$\theta_k \leftarrow g(p(\mathbf{z}),\ k)$

Clustering Assignment Estimation

Component Parameter Estimation

- How to keep track of convergence?
  - A simple rule for *k-means*

When the assignments don't change.

  - Alternatively keep track of the *k-means* global objective:

$$L(\Theta) = \sum_n \sum_k \left|\left|x_n - \theta_{z_n}\right|\right|^2$$

  - Dirichlet Processs Mixture with Variational Inference
    - Lower bound on the marginal likelihood

$$\mathcal{L} = h(\Theta, p(\mathbf{z}))$$

Loop until $\boxed{\mathcal{L}}$ convergence:

For $n = 1, \ldots, N$, and $k = 1, \ldots, K$

$$p(z_n = k) \leftarrow f(\Theta, \mathrm{k}, \mathrm{n})$$

For $k = 1, \ldots, K$

$$\theta_k \leftarrow g(p(\mathbf{z}),\ k)$$

$\boldsymbol{K \times N}$

$\boldsymbol{K}$

Clustering Assignment Estimation

Component Parameter Estimation

- What if the data doesn't fit in the disk?
- What if we want to accelerate this?

Divide the data into B batches

- Assumption:
  - Independently sampled assignment into batches
  - Enough samples inside each data batch
    - For latent components

$B \ll N$

$K \times N$

K

Loop until $\mathcal{L}$ convergence:

For $n = 1, \dots, N$, and $k = 1, \dots, K$

$p(z_n = k) \leftarrow f(\Theta, \mathrm{k}, \mathrm{n})$

For $k = 1, \dots, K$

$\theta_k \leftarrow g(p(\mathbf{z}),\ k)$

Clustering Assignment Estimation

Component Parameter Estimation

Divide the data into B batches

- Clusters are shared between data batches!

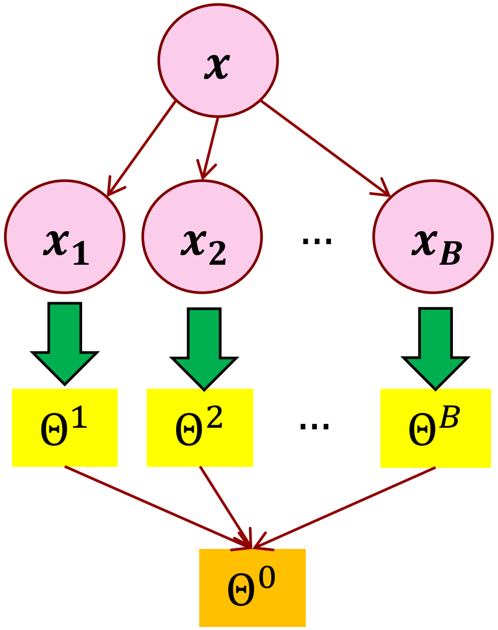
Define global / local cluster parameters

Global component parameters:

$$\Theta^0 = [\theta_1^0 \quad \theta_2^0 \quad \cdots \quad \theta_K^0]$$

Local component parameter:

$$\begin{aligned}
\Theta^1 &= [\theta_1^1 \quad \theta_2^1 \quad \cdots \quad \theta_K^1] \\
\Theta^2 &= [\theta_1^2 \quad \theta_2^2 \quad \cdots \quad \theta_K^2] \\
&\vdots \qquad\qquad \vdots \\
\Theta^B &= [\theta_1^B \quad \theta_2^B \quad \cdots \quad \theta_K^B]
\end{aligned}$$



$K \times N$

K

Loop until $\mathcal{L}$ convergence:

For $n = 1, \dots, N$, and $k = 1, \dots, K$

$p(z_n = k) \leftarrow f(\Theta, \mathrm{k}, \mathrm{n})$

For $k = 1, \dots, K$

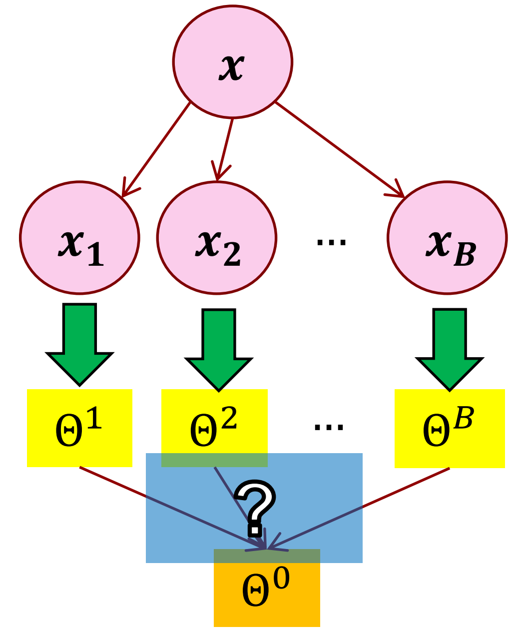
$\theta_k \leftarrow g(p(\mathbf{z}),\ k)$

Clustering Assignment Estimation

Component Parameter Estimation

- How to aggregate the parameters?

K-means example:
The global cluster center, is weighted average of the local cluster centers.



- Similar rules holds in DPM:
  - For each component: k $$\theta_k^0 = \sum_b \theta_k^{\mathrm{b}}$$
  - For all components: $$\Theta^0 = \sum_b \Theta^{\mathrm{b}}$$

$K \times N$

K

Loop until $\mathcal{L}$ convergence:

For $n = 1, \dots, N$, and $k = 1, \dots, K$

$p(z_n = k) \leftarrow f(\Theta, \mathrm{k}, \mathrm{n})$

For $k = 1, \dots, K$

$\theta_k \leftarrow g(p(\boldsymbol{z}),\ k)$

Clustering Assignment Estimation

Component Parameter Estimation

- How does the algorithm look like?

Loop until $\mathcal{L}$ convergence:

Randomly choose: $b \in \{1, 2, 3, \dots, B\}$

For $n \in \mathcal{B}_b$, and $k = 1, \dots, K$

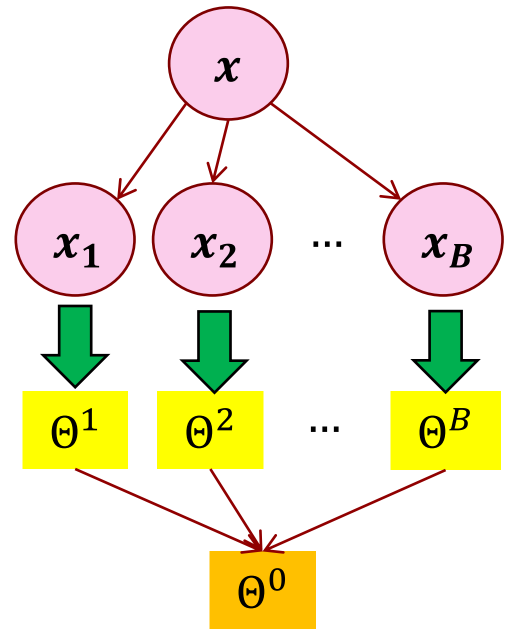
$p(z_n = k) \leftarrow f(\Theta^0, \mathrm{k}, \mathrm{n})$

For cluster $k = 1, 2, 3, \dots, K$

$\theta_k^{b\,(new)} \leftarrow g(p(\boldsymbol{z}),\ k, b)$

$\theta_k^0 \leftarrow \theta_k^0 - \theta_k^{b\,(old)} + \theta_k^{b\,(new)}$

$\theta_k^{b\,(old)} \leftarrow \theta_k^{b\,(new)}$



- Models and analysis for K-means:

Januzaj et al., “Towards effective and efficient distributed clustering”, ICDM, 2003

$K \times N$

K

Loop until $\mathcal{L}$ convergence:

For $n = 1, \dots, N$, and $k = 1, \dots, K$

$p(z_n = k) \leftarrow f(\Theta, \mathrm{k}, \mathrm{n})$

For $k = 1, \dots, K$

$\theta_k \leftarrow g(p(\mathbf{z}),\ k)$

Clustering Assignment Estimation

Component Parameter Estimation

- Compare these two:

(this work)

Loop until $\mathcal{L}$ (q) convergence:

Randomly choose: $b \in \{1, 2, 3, \dots, B\}$

For $n \in \mathcal{B}_b$, and $k = 1, \dots, K$

$p(z_n = k) \leftarrow f(\Theta^0, \mathrm{k}, \mathrm{n})$

For cluster $k = 1, 2, 3, \dots, K$

$\theta_k^{b\,(new)} \leftarrow g(p(\mathbf{z}),\ k, b)$

$\theta_k^0 \leftarrow \theta_k^0 - \theta_k^{b\,(old)} + \theta_k^{b\,(new)}$

$\theta_k^{b\,(old)} \leftarrow \theta_k^{b\,(new)}$

(Stochastic Optimization for DPM, Hoffman *et al.*, JMLR, 2013)

Loop until $\mathcal{L}$ (q) convergence:

Randomly choose: $b \in \{1, 2, 3, \dots, B\}$

For $n \in \mathcal{B}_b$, and $k = 1, \dots, K$

$p(z_n = k) \leftarrow f(\Theta^0, \mathrm{k}, \mathrm{n})$

For cluster $k = 1, 2, 3, \dots, K$

$\theta_k^{b} \leftarrow g(p(\mathbf{z}),\ k, b)$

$\theta_k^0 \leftarrow (1 - \rho_i)\theta_k^0 + \rho_i . \theta_k^b . \frac{n}{|\mathcal{B}_b|}$

$$\sum_i \rho_i \rightarrow +\infty,\ \sum_i \rho_i^2 < +\infty$$

$K \times N$

K

Loop until $\mathcal{L}$ convergence:

For $n = 1, \dots, N$, and $k = 1, \dots, K$
$$p(z_n = k) \leftarrow f(\Theta, \mathrm{k}, \mathrm{n})$$

For $k = 1, \dots, K$
$$\theta_k \leftarrow g(p(\mathbf{z}),\ k)$$

Clustering Assignment Estimation

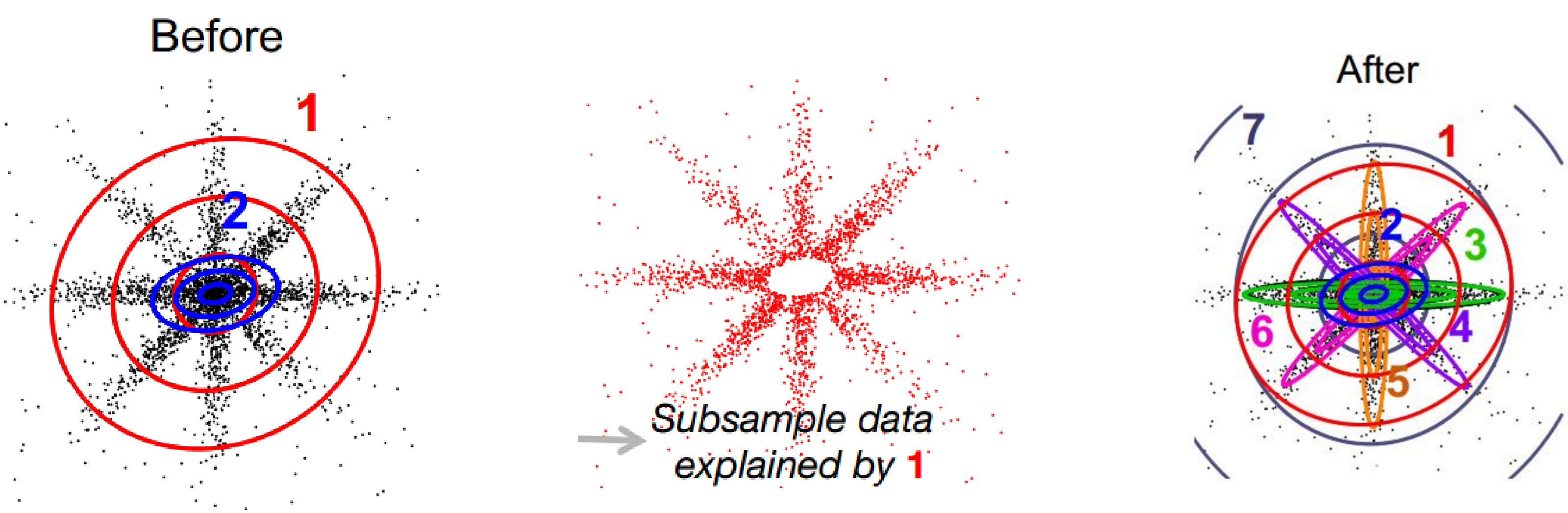
Component Parameter Estimation

Dirichlet Process Mixture (DPM)

- Note:
  - They use a nonparametric model!
  - But ....
    - the inference uses maximum-clusters
  - How to get adaptive number of maximum-clusters?
    - Heuristics to add new clusters, or remove them.

Birth moves

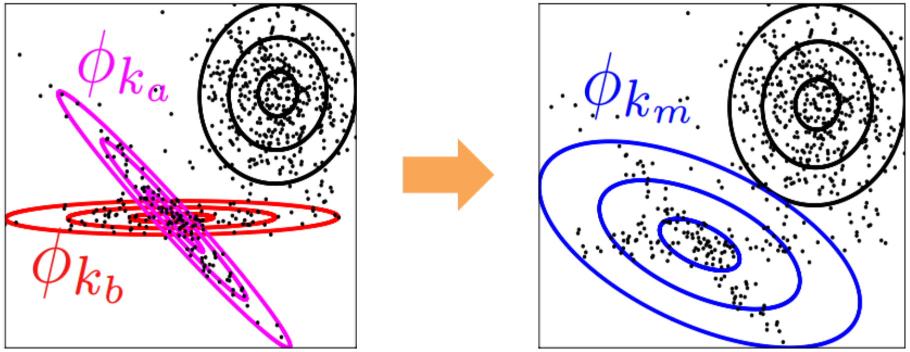
- The strategy in this work:
  - **Collection**:
    - Choose a random target component k'
    - Collect all the data points that $p(x_n = k') > \tau_{\text{threshold}}$ ($p(x_n = k') > \tau_{\text{threshold}}$)
  - **Creation:** run a DPM on the subsampled data ($K' = 10$)
  - **Adoption:** Update parameters with $K' + K$



Other birth moves?

- Past: split-merge schema for single-batch learning
  - E.g. EM (Ueda et al., 2000), Variational-HDP (Bryant and Sudderth, 2012), etc.
    - Split a new component
    - Fix everything
    - Run restricted updates.
    - Decide whether to keep it or not
  - Many similar Algorithms for k-means
    - (Hamerly & Elkan, NIPS, 2004), (Feng & Hammerly, NIPS, 2007), etc.
- This strategy unlikely to work in the batch mode:
  - Each batch might not contain enough examples of the missing component

Merge clusters



New cluster k_m takes over all responsibility of old clusters k_a and k_b:

$$\theta^0_{k_m} \leftarrow \theta^0_{k_a} + \theta^0_{k_b}$$

$$p(z_n = k_m) \leftarrow p(z_n = k_a) + p(z_n = k_b)$$

Accept or reject:

$$\mathcal{L}(q_{mrege}) > \mathcal{L}(q)?$$

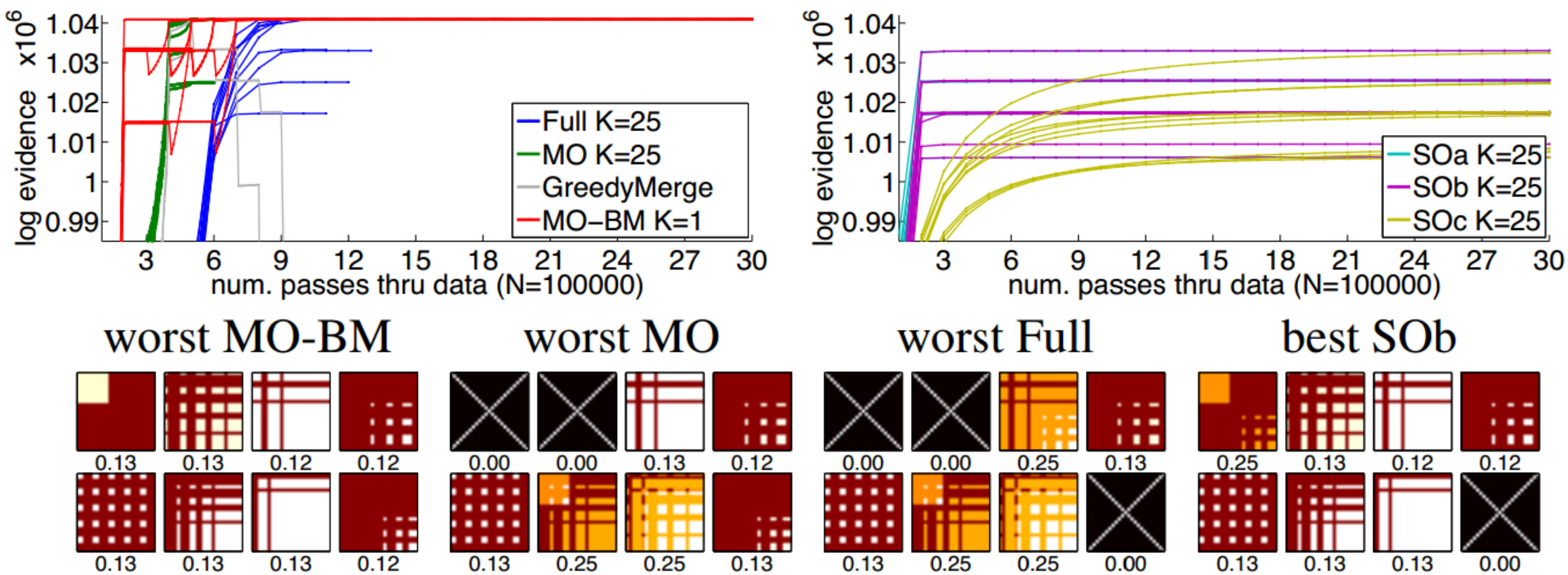
How to choose pair?

- Randomly select k_a
- Randomly select k_b proportional to the relative marginal likelihood:

$$p(k_b|k_a) \propto \frac{\mathcal{L}_{k_a+k_b}}{\mathcal{L}_{k_b}}$$

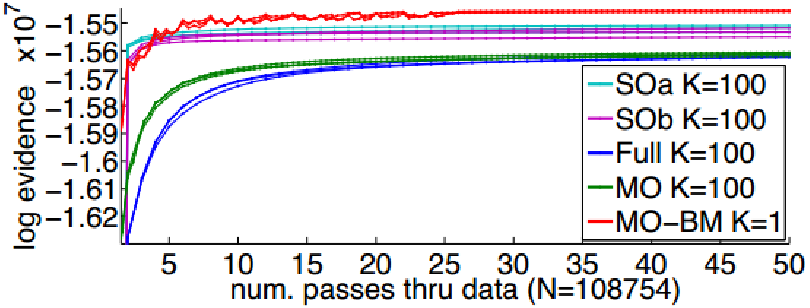
Results: toy data

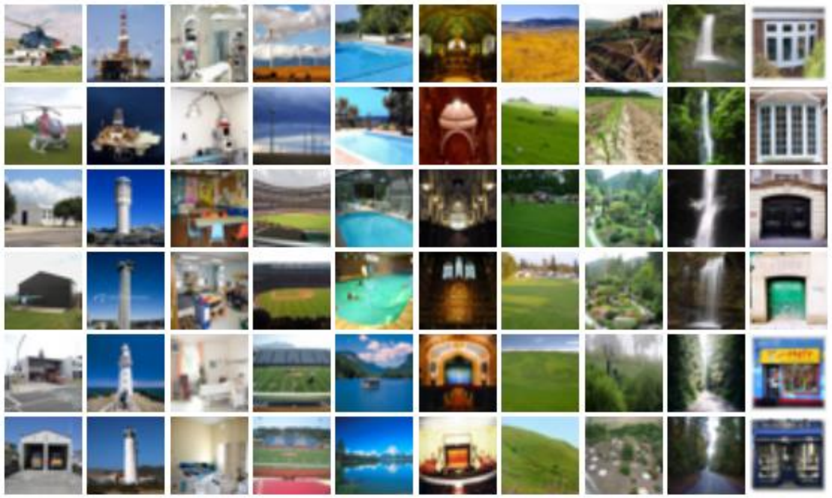
- Data (N=100000) synthetic image patches
- Generated by a zero mean GMM with 8 equally common components
- Each component has 25 × 25 covariance matrix producing 5 × 5 patches
- Goal: recovering these patches, and their size (K=8)
- B = 100 (1000 examples per batch)
- MO-BM starts with *K = 1,*
- Truncation-fixed start with K = 25 with 10 random initialization



Results: Clustering tiny images

- 108754 images of size 32 × 32
- Projected in 50 dimension using PCA
- MO-BM starting at *K = 1*, others have *K=100*
- full-mean DP-GMM





Summary

- A distributed algorithm for Dirichlet Process Mixture model
- Split-merge schema
- Interesting improvement over the similar methods for DPM.
- Theoretical convergence guarantees ?
- Theoretical justification for choosing batches B, or experiments investigating it?
- Previous "almost" similar algorithms, specially on *k-means ?*

Bayesian Inference

likelihood

Posterior

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{y})}$$

Prior

Marginal likelihood / Evidence

- Goal:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\boldsymbol{y})$$

- But posterior hard to calculate:

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})\boldsymbol{d\theta}}$$

Lower-bounding marginal likelihood

$$p(\boldsymbol{\theta}|\boldsymbol{x}) \sim q(\boldsymbol{\theta})$$

$$\log p(\boldsymbol{x}) \geq \log p(\boldsymbol{x}) - KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\boldsymbol{x}))$$
$$= \int q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta} = \mathcal{L}(q)$$

Given that,

$$KL(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\boldsymbol{x})) = \int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{x}|\boldsymbol{\theta})} d\boldsymbol{\theta}$$

Advantage

- Turn Bayesian inference into optimization
- Gives lower bound on the marginal likelihood

Disadvantage

- Add more non-convexity to the objective
- Cannot easily applied when non-conjugate family

$$g(\boldsymbol{\theta}) = p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

Variational Bayes for Conjugate families

- Given the joint distribution:

$$p(\boldsymbol{x}, \boldsymbol{\theta})$$

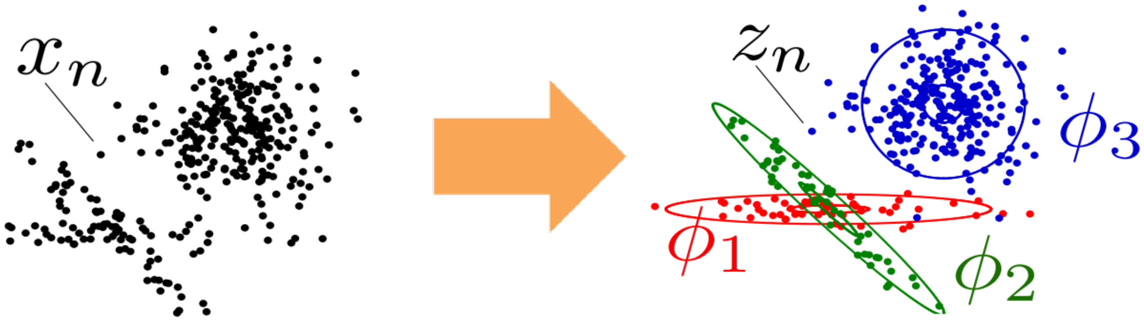
- And by making following decomposition assumption:

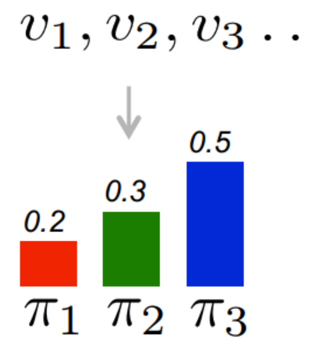
$$\boldsymbol{\theta} = [\theta_1, \dots, \theta_m], \qquad q(\theta_1, \dots, \theta_m) = \prod_{i=1}^{m} q(\theta_j)$$

- Optimal updates have the following form:

$$q(\theta_k) \propto \exp\left\{-\mathbb{E}_{q_{\setminus k}}[\log p\ (\boldsymbol{x}, \boldsymbol{\theta})]\right\}$$

Dirichlet Process (Stick Breaking)

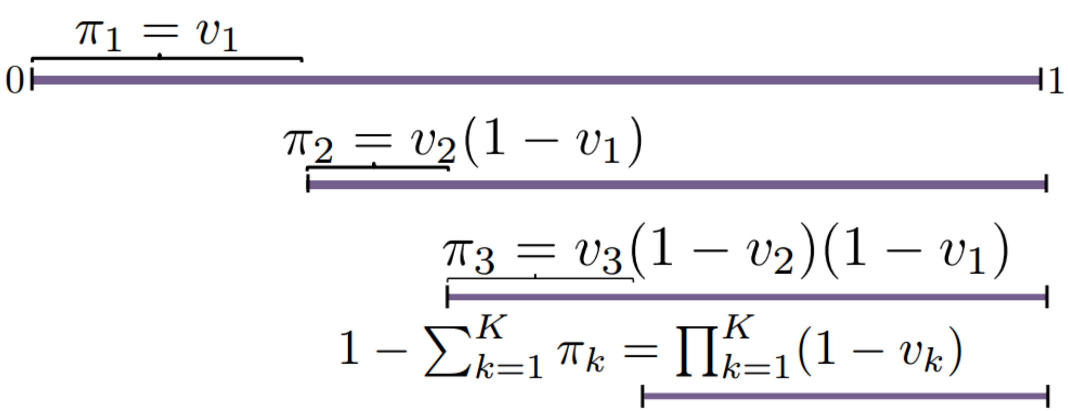




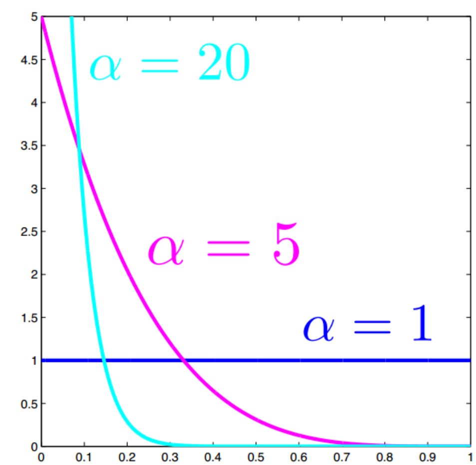
For each cluster $k = 1, 2, 3, \ldots$

- Cluster shape: $\phi_k \sim H(\lambda_0)$
- Stick proportion: $v_k \sim Beta(1, \alpha)$
- Cluster coefficient: $\pi_k = v_k \prod_{l=1}^{k}(1 - v_l)$

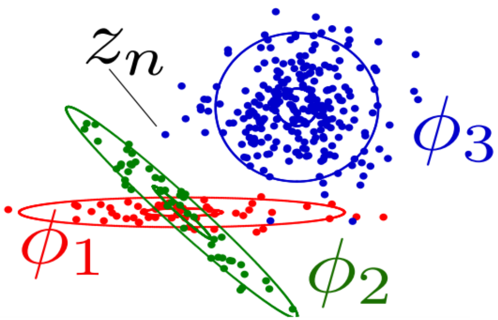
$\Big\} \pi \sim Stick(\alpha)$

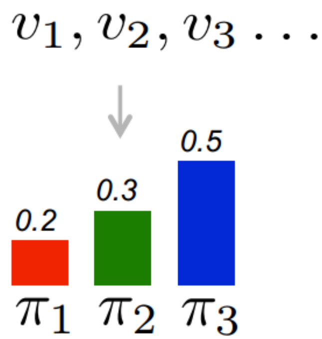


Stick-breaking (Sethuraman,1994)



Dirichlet Process Mixture model



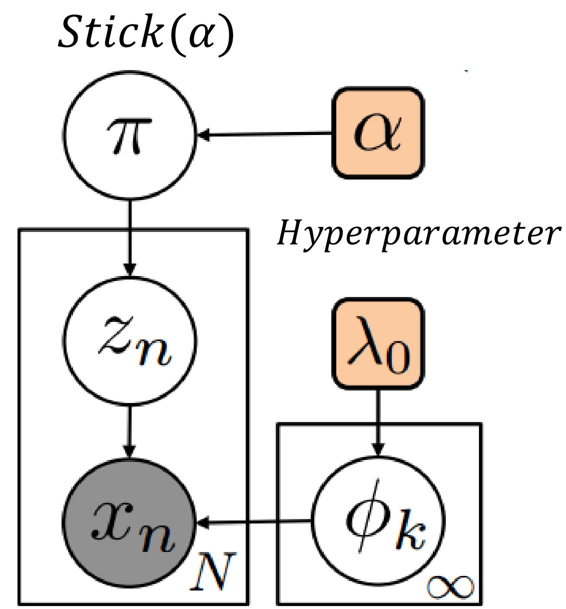


For each cluster $k = 1, 2, 3, \dots$

- Cluster shape: $\phi_k \sim H(\lambda_0)$
- Stick proportion: $v_k \sim Beta(1, \alpha)$
- Cluster coefficient: $\pi_k = v_k \prod_{l=1}^{k}(1 - v_l)$

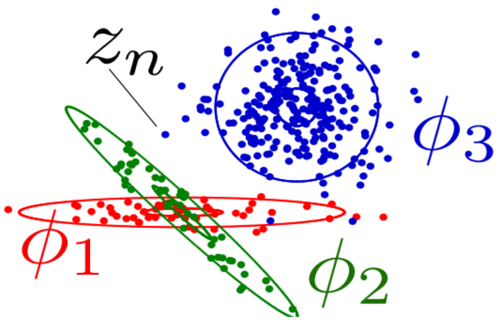
For each data point: $n = 1, 2, 3, \dots$

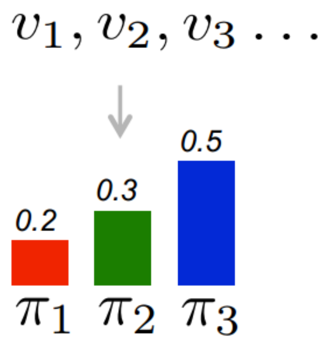
- Cluster assignment: $z_n \sim Cat(\pi)$
- Observation: $x_n \sim \phi_{z_n}$

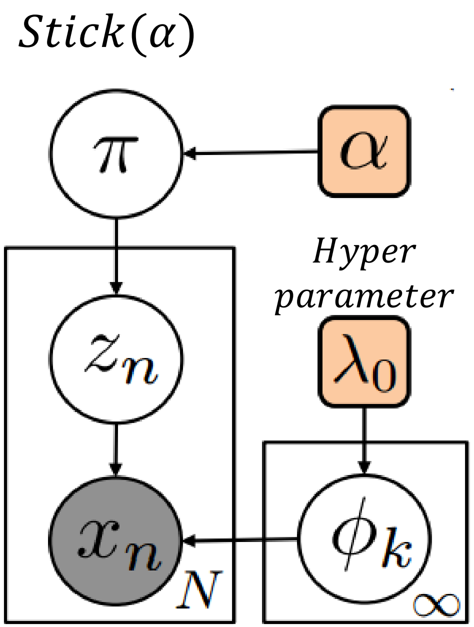


Posterior variables: $\Theta = \{z_n, v_k, \phi_k\}$
Approximation: $q(\mathrm{z_n}, v_k, \phi_k)$

Dirichlet Process Mixture model







For each data point n and clusters k

- $q(z_n = k) = r_{nk} \propto \exp\{\mathbb{E}_q[\log \pi_k(v) + \log p(x_n|\phi_k)\ ]\}$

For cluster $k = 1, 2, 3, \ldots, K$

- $N_k^0 \leftarrow \sum_n r_{nk}$
- $s_k^0 \leftarrow \sum_{n=1}^{N} r_{nk} t(x_n)$
- $\lambda_k \leftarrow \lambda_0 + s_k^0$

For cluster $k = 1, 2, 3, \ldots, K$

- $\alpha_k^0 \leftarrow 1 + N_k^0$
- $\alpha_k^0 \leftarrow \alpha + \sum_{l>k} N_l^0$

Stochastic Variational Bayes

Hoffman et al., JMLR, 2013

Stochastically divide data into B batches:

$$\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_B$$

- For each batch: $b = 1, 2, 3, \dots, B$
  - $r \leftarrow EStep(\mathcal{B}_b, \alpha, \lambda)$
  - For each cluster $k = 1, 2, 3, \dots, K$
    - $s_k^b \leftarrow \sum_{n \in \mathcal{B}_b} r_{nk} \quad t(x_n)$
    - $\lambda_k^b \leftarrow \lambda_0 + \frac{N}{|\mathcal{B}_b|} s_k^b$
    - $\lambda_k \leftarrow \rho_t \lambda_k^b + (1 - \rho_t)\lambda_k$
    - Similarly for stick weights

Convergence condition on ρ_t

$$\sum_t \rho_t \to \infty, \sum_t \rho_t^2 < \infty$$

Memoized Variational Bayes

Hughes & Sudderth, NIPS 2013

Stochastically divide data into B batches:

$$\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_B$$

- For each batch: $b = 1, 2, 3, \ldots, B$
  - $r \leftarrow EStep(\mathcal{B}_b, \alpha, \lambda)$
  - For data item $k = 1, 2, 3, \ldots, K$
    - $s_k^0 \leftarrow s_k^0 - s_k^b$
    - $s_k^b \leftarrow \sum_{n \in \mathcal{B}_b} r_{nk} t(x_n)$
    - $s_k^0 \leftarrow s_k^0 + s_k^b$
    - $\lambda_k \leftarrow \lambda_0 + s_k^0$

Global variables:

$$\begin{matrix} s_1^0 & s_2^0 & \cdots & s_K^0 \end{matrix}$$

$$s_k^0 = \sum_b s_k^b$$

Local variables:

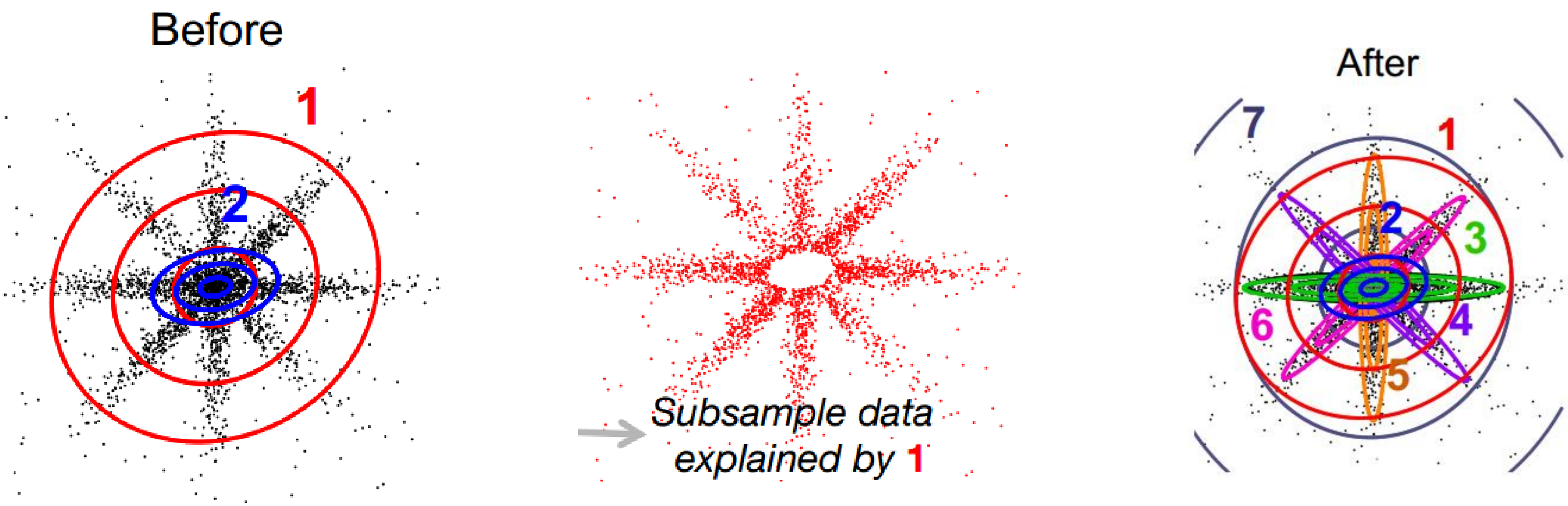
$$\begin{matrix} s_1^1 & s_2^1 & \cdots & s_K^1 \\ s_1^2 & s_2^2 & \cdots & s_K^2 \\ \vdots & \vdots & \ddots & \vdots \\ s_1^B & s_2^B & \cdots & s_K^B \end{matrix}$$

Birth moves

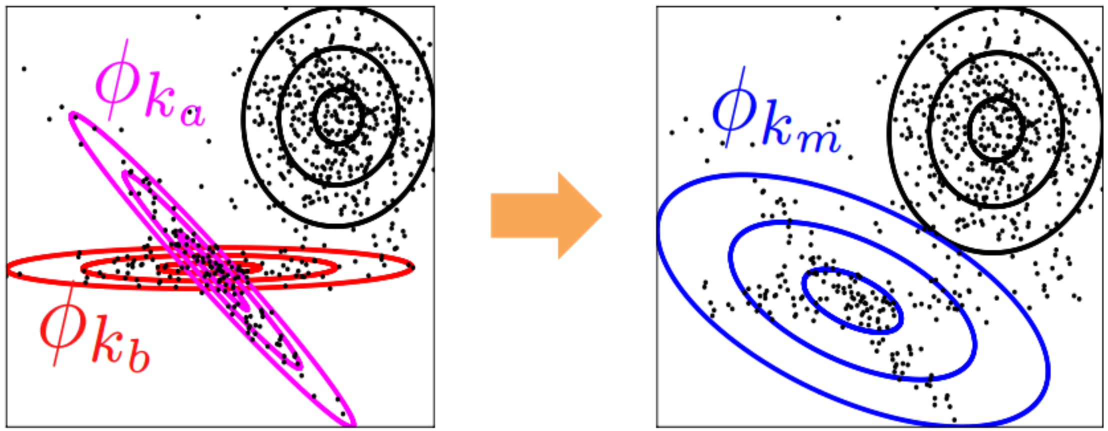
- Conventional variatioanl approximation:
  - Truncation on the number of components
- Need to have an adaptive way to add new components
- Past: split-merge schema for single-batch learning
  - E.g. EM (Ueda et al., 2000), Variational-HDP (Bryant and Sudderth, 2012), etc.
  - Split a new component
  - Fix everything
  - Run restricted updates.
  - Decide whether to keep it or not
- This strategy unlikely to work in the batch mode:
  - Each batch might not contain enough examples of the missing component

Birth moves

- The strategy in this work:
  - **Collection**: subsample data in the targeted component k'
  - **Creation:** run a DPM on the subsampled data ($K' = 10$)
  - **Adoption:** Update parameters with $K' + K$



Merge clusters



New cluster k_m takes over all responsibility of old clusters k_a and k_b:

$r_{nk_m} \leftarrow r_{nk_a} + r_{nk_b}$ $\qquad$ $N^0_{k_m} \leftarrow N^0_{k_a} + N^0_{k_b}$ $\qquad$ $s^0_{k_m} \leftarrow s^0_{k_a} + s^0_{k_b}$

Accept or reject:

$$\mathcal{L}(q_{mrege}) > \mathcal{L}(q)?$$

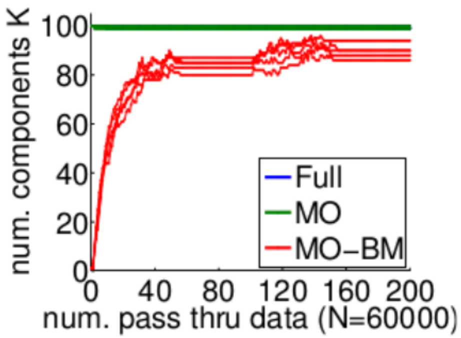
How to choose pair?

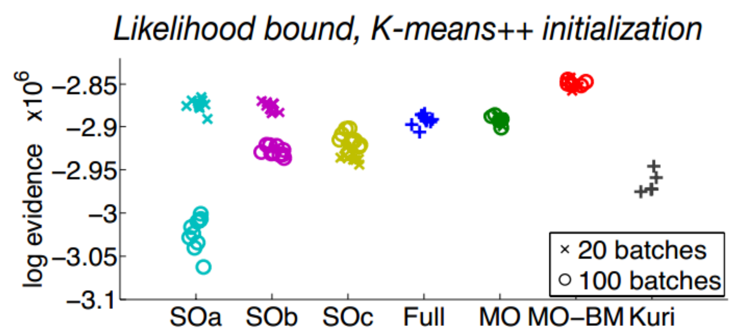
Randomly sample proportional to the relative marginal likelihood:

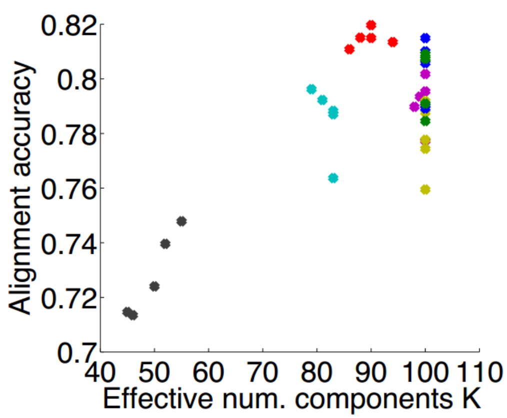
$$\frac{M(S_{k_a} + S_{k_b})}{M(S_{k_a}) + M(S_{k_b})}$$

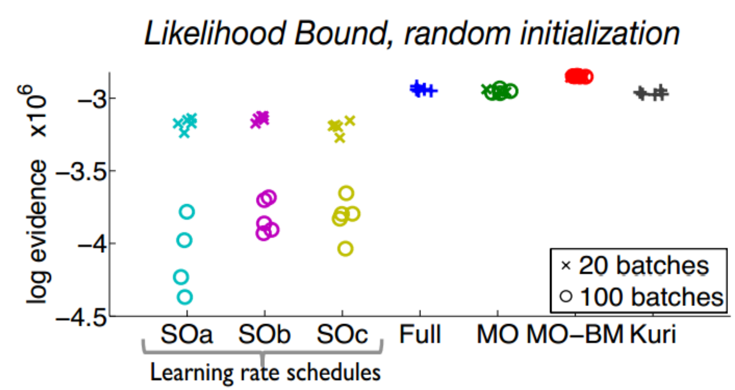
Results: Clustering Handwritten digits

- Clustering N = 60000 MNIST images of handwritten digits 0-9.
- As preprocessing, all images projected to D = 50 via PCA.









Kuri: Kurihara et al. "Accelerated variational ...", NIPS 2006

References

- Michael C. Hughes, and Erik Sudderth. "Memoized Online Variational Inference for Dirichlet Process Mixture Models." *Advances in Neural Information Processing Systems*. 2013.
- Erik Sudderth slides: http://cs.brown.edu/~sudderth/slides/isba14variationalHDP.pdf
- Kyle Ulrich slides: http://people.ee.duke.edu/~lcarin/Kyle6.27.2014.pdf