Mixture of PPCA

Minh, 21/01/2019

Mixture of PPCA

Motivation:

- Mixture model is a tool for soft-clustering, outputs the clustering assignments for each datapoint, but we can not visualize the groups of datapoints in 2D.
- PPCA can reduce the dimensions of data, but we still use the supervised labels to color the points in the visualization.
- The group of points in 2D $\neq$ cluster of points in high dim. (HD)
- Mixture of PPCA model can infer:
  - the 2D positions for each data point
  - and their clustering assignment
- The expected visualization with MPPCA:
  - the points of the same cluster in HD are placed close together and/or in the same group in 2D.
  - multi-views visualization: having multiple visualizations for each component
    $\rightarrow$ discover the local structure of each component.

Input:

- Observed data: $\mathbf{X} = \{\mathbf{x}_i\}, \quad \mathbf{x}_i \in \mathbb{R}^D, \quad i = 1, .., N$

Output:

- Latent position in 2D: $\mathbf{Z} = \{\mathbf{z}_i\}, \quad \mathbf{z}_i \in \mathbb{R}^M, \quad i = 1, .., N$
- Latent clustering assignment: $\mathbf{G} = \{g_{ik}\}, \quad g_{ik} \in \{0, 1\}, \quad k = 1, .., K$ ($g_{ik} = 1$ iif $\mathbf{x}_i$ belongs to the k^{th} component)

Evaluation:

- Visualization quality (hard to measure)
- Clustering quality (e.g. VMeasure)

Traditional mixture of Gaussians

- A discrete indicator variable $g_{ik} \in \{0, 1\}$ indicates whether the k^{th} component generates the datapoint $\mathbf{x}_i$:

$$p(\mathbf{x}_i \mid g_{ik} = 1) = \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \sigma_k \mathbf{I}_D)$$

where each component is an isotropic multivariate gaussian with mean $\boldsymbol{\mu}_k \in \mathbb{R}^D$ and scalar variance σ_k.

- By summing all possible assignment states of each point, its marginal distribution is:

$$p(\mathbf{x}_i) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \sigma_k \mathbf{I}_D)$$

where the mixing coefficient π_k represents the contributon of the k^{th} component.

Mixture of PPCA

For each component, replace the Gaussian distribution by a PPCA distribution: (note that the PPCA distribution is conditioned on the latent variable $\mathbf{z}_i$)

$$p(\mathbf{x}_i \mid \mathbf{z}_i) = \sum_{k=1}^{K} \pi_k \underbrace{\mathrm{PPCA}(\mathbf{x}_i \mid \mathbf{z}_i, k)}_{\text{in fact, is also a Gaussian}}$$

Generative process

Each point $\mathbf{x}_i \in \mathbb{R}^D$ is generated from a corresponding latent variable $\mathbf{z}_i \in \mathbb{R}^M$ as following:

1. Choose one of K components from which $\mathbf{x}_i$ will be generated.
2. Sample a latent variable $\mathbf{z}_i$ from an unit gaussian.
3. $\mathbf{x}_i$ is mapped from $\mathbf{z}_i$ via a projection matrix $\mathbf{W}_K \in \mathbb{R}^{D \times M}$, then shifted to the center $\boldsymbol{\mu}_k$ and disturbed by a variance σ_k.

   i.e., $\mathbf{x}_i$ is sampled from $\mathrm{PPCA}(\mathbf{x}_i \mid k) \equiv \mathcal{N}(\boldsymbol{\mu}_k + \mathbf{z}_i \mathbf{W}_k^T, \sigma_k \mathbf{I}_D)$

Elements of MPPCA model (1)

Parameters and their priors

- Global parameter: $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_K]$

$$\boldsymbol{\pi} \sim \text{Dirichlet}(\alpha_0), \qquad \alpha_0 = \overbrace{\left[\frac{1}{K}, \ldots, \frac{1}{K}\right]}^{K \text{elements}}$$

- Local parameters for each components:

$$\boldsymbol{\mu}_k \sim \mathcal{N}(\mathbf{0}_D, \mathbf{I}_D)$$

$$\sigma_k \sim \text{LogNormal}(\sigma_k \mid 0, 1)$$

$$\mathbf{W}_k^{(j)} \sim \mathcal{N}(\mathbf{W}_k^{(j)} \mid \mathbf{0}_M, \mathbf{I}_M)$$

where $\mathbf{W}_k^{(j)}$ is a row of the projection matrix $\mathbf{W}_k = \begin{bmatrix} \mathbf{W}_k^{(1)} \\ \vdots \\ \mathbf{W}_k^{(D)} \end{bmatrix}$

Elements of MPPCA model (2)

Latent variables and their priors

Latent variables for each datapoint:

- Latent position in low dimensional space:

$$\mathbf{z}_i \sim \mathcal{N}(\mathbf{z}_i \mid \mathbf{0}_D, \mathbf{I}_D)$$

- Latent clustering assignment:

The cluster id for each $\mathbf{x}_i$ is assigned exactly one of K components (categories): $\mathbf{g}_i = \underbrace{[0, \ldots, 1, \ldots, 0]}_{K elements}$.

Thus the clustering assignment is modeled by the discrete Categorical distribution:

$$\mathbf{g}_i \sim \text{Categorical}(\text{probs} = [\pi_1, \ldots, \pi_K])$$

Elements of MPPCA model (3)

Observed variables

Based on the clustering assigment of each point $\mathbf{g}_i$, we determine the component k to which the datapoint $\mathbf{x}_i$ belongs:

$$\mathbf{x}_i \mid \mathbf{z}_i, \mathbf{g}_i \sim \mathrm{PPCA}(\mathbf{x}_i \mid \mathbf{z}_i, k)$$

Thus we have

$$p(\mathbf{x}_i \mid \mathbf{z}_i, \mathbf{g}_i) = \mathcal{N}(\boldsymbol{\mu}_k + \mathbf{z}_i \mathbf{W}_k^T, \sigma_k \mathbf{I}_D)$$

Marginalize out the discrete latent variable $\mathbf{g}_i$:

$$\begin{aligned} p(\mathbf{x}_i \mid \mathbf{z}_i) &= \sum_g p(\mathbf{x}_i \mid \mathbf{z}_i, \mathbf{g}_i) \\ &= \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{\mu}_k + \mathbf{z}_i \mathbf{W}_k^T, \sigma_k \mathbf{I}_D) \end{aligned}$$

The model

Likelihood of each observation

$$p(\mathbf{x}_i \mid \mathbf{z}_i, \boldsymbol{\mu}_k, \sigma_k, \mathbf{W}_k) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{\mu}_k + \mathbf{z}_i \mathbf{W}_k^T, \sigma_k \mathbf{I}_D)$$

The local parameters are denoted ensemble as $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k, \mathbf{W}_k\}$

Thus the likelihood for each observation is

$$p(\mathbf{x}_i \mid \mathbf{z}_i, \boldsymbol{\theta}_k)$$

Denote $\boldsymbol{\theta} = \{\boldsymbol{\theta}_k\}_{k=1}^{K}$.

The likelihood for all dataset is $p_{\boldsymbol{\theta}}(\mathbf{X} \mid \mathbf{Z})$.

The prior of latent variables is $p_{\boldsymbol{\theta}}(\mathbf{Z})$.

The posterior that we have to infer is $p_{\boldsymbol{\theta}}(\mathbf{Z} \mid \mathbf{X})$.

Posterior distribution

The true posterior $p_{\boldsymbol{\theta}}(\mathbf{Z} \mid \mathbf{X})$ will be approximated by the variational distribution $q_{\boldsymbol{\phi}}(\mathbf{Z})$.

The goal is thus to find the variational parameters $\boldsymbol{\phi}$ which make $q_{\boldsymbol{\phi}}(\mathbf{Z})$ close to $p_{\boldsymbol{\theta}}(\mathbf{Z} \mid \mathbf{X})$ as much as possible.

The log evidence can be written as

$$\log p_\theta(\mathbf{X}) = \mathrm{ELBO} + \mathrm{KL}\left(\mathrm{q}_\phi(\mathbf{Z}) \| \mathrm{p}_\theta(\mathbf{Z}|\mathbf{X})\right)$$

where the Evidence Lower BOund is defined as:

$$\mathrm{ELBO} \equiv \mathbb{E}_{q_\phi(\mathbf{Z})}\left[\log p_\theta(\mathbf{X}, \mathbf{Z}) - \log q_\phi(\mathbf{Z})\right]$$

Since $\log p_\theta(\mathbf{X})$ is constant. The goal of maximize the $\mathrm{KL}\left(\mathrm{q}_\phi(\mathbf{Z}) \| \mathrm{p}_\theta(\mathbf{Z}|\mathbf{X})\right)$ is equivalent to minimize the ELBO.

Implementation (1)

The pyro library supports to minimize the ELBO w.r.t the variational parameters ϕ (and the model parameters θ).

The following gradient step is calculated automatically by an optimizer like Adam.

$$\nabla_{\theta,\phi}\mathrm{ELBO} = \nabla_{\theta,\phi}\mathbb{E}_{q_\phi(\mathbf{Z})}\left[\log p_\theta(\mathbf{X},\mathbf{Z}) - \log q_\phi(\mathbf{Z})\right]$$

We have to define:

- the distributions over the latent variables, the params and the observed data (called `model`):

  $p(\boldsymbol{\pi}), p(\boldsymbol{\mu}), p(\boldsymbol{\sigma}), p(\mathbf{W}), p(\mathbf{Z}), p(\mathbf{G})$ and
  $p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\theta}) \equiv p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \mathbf{W})$.

- the variational distributions (called `guide`):

  $q(\boldsymbol{\pi}), q(\boldsymbol{\mu}), q(\boldsymbol{\sigma}), q(\mathbf{W}), q(\mathbf{Z}), q(\mathbf{G})$

Implementation (2)

The choice of `guide`

- MAP estimation for the parameters and fully Baysiean inference for the latent variables:

$$
\begin{aligned}
q(\boldsymbol{\pi}) &\sim Delta(\boldsymbol{\pi}) \\
q(\boldsymbol{\mu}) &\sim Delta(\boldsymbol{\mu}) \\
q(\boldsymbol{\sigma}) &\sim Delta(\boldsymbol{\sigma}) \\
q(\mathbf{W}) &\sim Delta(\mathbf{W}) \\
q(\mathbf{Z}) &\sim Normal(\mathbf{Z}) \\
q(\mathbf{G}) &\sim Categorical(\mathbf{G})
\end{aligned}
$$

- Fully Bayesian inference:

  Since Delta distribution is used as point-estimate for each parameter.
  If we replace the Delta distribution by Normal distributions, we have fully Bayesian inference.

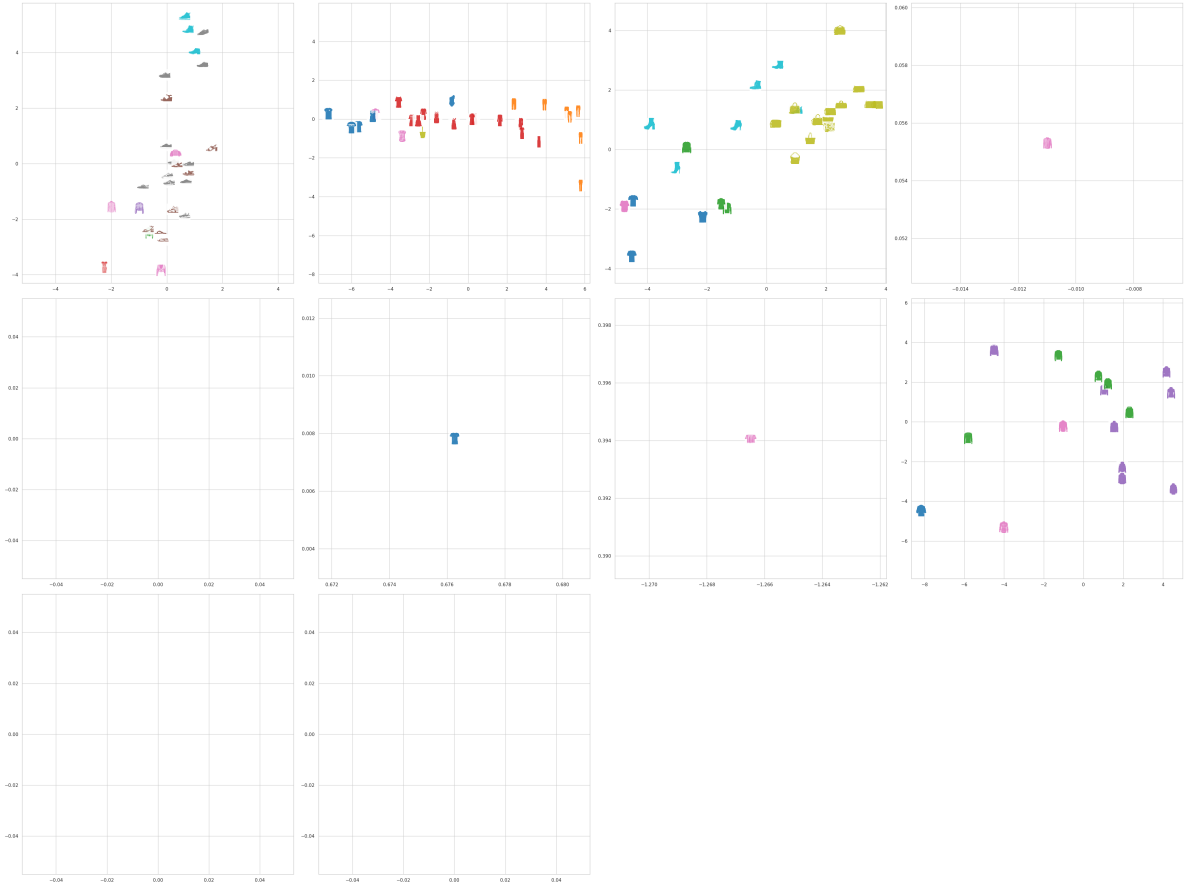
Integrate the user constraints

- The user constraint is in form: `a point` i `should belong to cluster` k.
- To force the cluster assignment $\mathbf{g}_i = [0, \ldots, 1, \ldots, 0]$ for the interacted point j, define a promoted mixing coefficient that encourages the k^{th} components:

  $\boldsymbol{\pi}_j = [\text{small number}, \ldots, \text{large number} \ldots, \text{small number}]$

- All other uninteracted points share the same global $\boldsymbol{\pi}$

Current result of non-interactive MPPCA model



Using Delta distribution for q, dataset `FASHION100`, `K = 10`, `vmeasure~0.51`