Machine Learning

Week 6: Unsupervised Learning: PCA, Mixture Models, Cluster Analysis

Mahesan Niranjan

School of Electronics and Computer Science
University of Southampton

Autumn Semester 2016/17

Week Six: Overview

- Review: Maximum Likelihood and Bayesian Estimation
- Deriving Principal Component Analysis
- Mixture Gaussian Model
- Expectation Maximization (EM) Algorithm
- K-Means Clustering

Note:

> You need not learn the derivations in estimating mixture model parameters by heart. But we need to go through the algebra to gain an insight into the formal basis of the algorithms we use in Machine Learning.

Unsupervised Learning

- Given: $\{\boldsymbol{x}_n\}_{n=1}^N$ (as opposed to $\{\boldsymbol{x}_n, f_n\}_{n=1}^N$)
- We might extract cluster structures
  - Notion of distance between points of data
  - Criterion to determine how many clusters (often from prior knowledge)
  - Underlying probabilistic model
- We might project data onto a subspace $\boldsymbol{x}_n \in \mathcal{R}^d \longrightarrow \boldsymbol{y}_n \in \mathcal{R}^q$
  - $q = 2$ helps visualization
  - Subspace representation useful for
    - Data compression
    - Sometimes used to reduce features

Semi Supervised Learning:

$$\{\boldsymbol{x}_n, f_n\}_{n=1}^{N_1} \text{ and } \{\boldsymbol{x}_n\}_{n=N_1+1}^{N_2}$$

Constrained Optimization: Lagrange Multipliers

Problem:

- Maximize $f(\boldsymbol{x})$ (with respect to $\boldsymbol{x}$)
- Subject to $g(\boldsymbol{x}) = c$

Method:

- Construct a function

$$L(\boldsymbol{x}, \lambda) = f(\boldsymbol{x}) - \lambda (g(\boldsymbol{x}) - c)$$

- L is called a Lagrangian; λ is called a Lagrange Multiplier
- The problem now is an unconstrained problem; we look for turning points by

$$\frac{\partial L(\boldsymbol{x}, \lambda)}{\partial \boldsymbol{x}} = \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} - \lambda \frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}} = 0$$

Example of Lagrange Multipliers:
Principal Component Analysis

- N data $\boldsymbol{x}_n \in \mathcal{R}^d$ distributed with mean $\boldsymbol{m}$ and covariance matrix $\boldsymbol{C}$.
- Project onto direction $\boldsymbol{u}$; find the direction that maximizes projected variance.
- Projected variance is $\boldsymbol{u}^t \boldsymbol{C} \boldsymbol{u}$
- We are only interested in the direction; not in increasing the projected variance by choosing $\boldsymbol{u}$ with large magnitude.
- Set up a constrained optimization problem

$$\max_{\boldsymbol{u}} \boldsymbol{u}^t \boldsymbol{C} \boldsymbol{u} \quad \text{subject to } \boldsymbol{u}^t \boldsymbol{u} = 1$$

- Lagrangian

$$\mathcal{L} = \boldsymbol{u}^t \boldsymbol{C} \boldsymbol{u} - \lambda \left[\boldsymbol{u}^t \boldsymbol{u} - 1 \right]$$

- $\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}} = 0 \implies \boldsymbol{C}\boldsymbol{u} = \lambda \boldsymbol{u}$; *i.e.* principal directions are eigenvectors of covariance

Mixture Model

We write a mixture of Gaussian densities:

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

- If the mixing proportions π_k satisfy
  - $\pi_k \geq 0$
  - $\sum_{k=1}^{K} \pi_k = 1$

  $p(\boldsymbol{x})$ is a proper probability density.

- More powerful model – useful when data is multi-modal
- Parameters are: proportions, means and covariance matrices
- Parameter estimation $(\pi_k, \boldsymbol{\mu}_k \boldsymbol{\Sigma}_k)$ is not easy.
- z_{nk}: association of n^{th} data to k^{th} mode unknown (latent)

Log Likelihood:

(Δ represents all the means and covariances and $\boldsymbol{\pi}$ is a vector holding all the π_i's)

$$\begin{aligned}\mathcal{L} &= \log p\left(\boldsymbol{X}|\Delta,\boldsymbol{\pi}\right)\\ &= \sum_{n=1}^{N}\log\sum_{k=1}^{K}\pi_k p\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)\end{aligned}$$

Note log of sums of variables; inconvenient to work with.

Jensen's Inequality

$$\log E_{p(z)}\left\{f(z)\right\} \geq E_{p(z)}\left\{\log f(z)\right\}$$

- Introduce a new variable q_{nk}: $q_{nk} \geq 0$ and $\sum_{k=1}^{K} q_{nk} = 1$
  At every data n, we are defining a new probability distribution over the K components of the mixture model.
- We multiply and divide by the new variable:

$$\mathcal{L} = \sum_{n=1}^{N}\log\sum_{k=1}^{K}\pi_k p\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)\frac{q_{nk}}{q_{nk}}$$

We now treat the weighted sum as expectation over the newly introduced distribution:

$$\begin{aligned}\mathcal{L} &= \sum_{n=1}^{N}\log\sum_{k=1}^{K}q_{nk}\frac{\pi_k p\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)}{q_{nk}}\\ &= \sum_{n=1}^{N}\log\mathbf{E}_{q_{nk}}\left\{\frac{\pi_k p\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)}{q_{nk}}\right\}\end{aligned}$$

That gives a form in which Jensen's inequality may be applied.

$$\begin{aligned}\mathcal{L} &= \sum_{n=1}^{N}\log\mathbf{E}_{q_{nk}}\left\{\frac{\pi_k p\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)}{q_{nk}}\right\}\\ &\geq \sum_{n=1}^{N}\mathbf{E}_{q_{nk}}\left\{\log\frac{\pi_k p\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)}{q_{nk}}\right\}\end{aligned}$$

What we do is to optimize this lower bound, rather than the log likelihood itself, with respect to the unknowns $\{q_{nk}, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$.

Mixture Model (cont'd)

$$\begin{aligned}\mathcal{B} &= \sum_{n=1}^{N} \mathbf{E}_{q_{nk}} \left\{ \log \frac{\pi_k p(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{q_{nk}} \right\} \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} q_{nk} \log \left(\frac{\pi_k p(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{q_{nk}} \right) \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} q_{nk} \log \pi_k + \sum_{n=1}^{N}\sum_{k=1}^{K} q_{nk} \log p(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) - \sum_{n=1}^{N}\sum_{k=1}^{K} q_{nk} \log q_{nk}\end{aligned}$$

The task now is to maximize this ($\mathcal{B}$) with respect to the unknowns.

Maximize with respect to π_k

- Only the first term depends on π_k
- But we need to constrain the solutions for π_k because $\sum_{k=1}^{K} \pi_k = 1$.
- Set up the Lagrangian:

$$B_1 = \sum_{n=1}^{N}\sum_{k=1}^{K} q_{nk} \log \pi_k - \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

- Differentiate and equate to zero:

$$\frac{\partial B_1}{\partial \pi_k} = \frac{\sum_{n=1}^{N} q_{nk}}{\pi_k} - \lambda = 0$$

$$\sum_{n=1}^{N} q_{nk} = \lambda \pi_k$$

Sum both sides over k

$$\begin{aligned}\sum_{k=1}^{K}\sum_{n=1}^{N} q_{nk} &= \lambda \sum_{k=1}^{K} \pi_k \\ N &= \lambda\end{aligned}$$

Hence $\pi_k = \frac{1}{N} \sum_{n=1}^{N} q_{nk}$

Maximizing with respect to $\boldsymbol{\mu}_k$

- Only the second term of the bound $\mathcal{B}$ depends on $\boldsymbol{\mu}_k$

$$\begin{aligned} B_2 &= \sum_{n=1}^{N}\sum_{k=1}^{K} q_{nk} \log\left(\frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^t \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)\right)\right) \\ &= -\frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} q_{nk} \log\left((2\pi)^{d/2}|\boldsymbol{\Sigma}_k|\right) - \frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^t \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k) \end{aligned}$$

- Differentiate:

$$\frac{\partial B_2}{\partial \boldsymbol{\mu}_k} = \sum_{n=1}^{N} q_{nk} \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_n - \boldsymbol{\mu}_k).$$

- Equate to zero and re-arrange terms

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} q_{nk} \boldsymbol{x}_n}{\sum_{n=1}^{N} q_{nk}}$$

Maximizing with respect to $\boldsymbol{\Sigma}_k$

- Again only the second term matters; differentiating with respect to $\boldsymbol{\Sigma}_k$ is tricky (we'll not do this)
- Answer

$$\boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^{N} q_{nk}(\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^t}{\sum_{n=1}^{N} q_{nk}}$$

Updating q_{nk} needs to recognize the constraints (sum to one)

$$B = \sum_{n=1}^{N}\sum_{k=1}^{K} q_{nk}\log\pi_k + \sum_{n=1}^{N}\sum_{k=1}^{K} q_{nk}\log p(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k) - \sum_{n=1}^{N}\sum_{k=1}^{K} q_{nk}\log q_{nk} - \lambda\left(\sum_{k=1}^{K} q_{nk} - 1\right)$$

$$\frac{\partial B}{\partial q_{nk}} = \log\pi_k + \log p\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right) - (1+\log q_{nk}) - \lambda$$

$$\begin{aligned} 1+\log q_{nk}+\lambda &= \log\pi_k + logp\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right) \\ \exp\left(\log q_{nk} + (\lambda+1)\right) &= \exp\left(\log\pi_k + \log p\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)\right) \\ q_{nk}\exp(\lambda+1) &= \pi_k p\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right) \end{aligned}$$

Sum over mixture components to get the Lagrange multiplier.

$$\exp(\lambda+1)\sum_{k=1}^{K} = \sum_{k=1}^{K}\pi_k p\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)$$

Because q_{nk} should sum to one, we have

$$q_{nk} = \frac{\pi_k p\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K}\pi_j p\left(\boldsymbol{x}_n|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j\right)}$$

Summary of Algorithm

$$\begin{aligned} \pi_k &= \frac{1}{N}\sum_{n=1}^{N} q_{nk} \\ \boldsymbol{\mu}_k &= \frac{\sum_{n=1}^{N} q_{nk}\boldsymbol{x}_n}{\sum_{n=1}^{N} q_{nk}} \\ \boldsymbol{\Sigma}_k &= \frac{\sum_{n=1}^{N} q_{nk}(\boldsymbol{x}_n-\boldsymbol{\mu}_k)(\boldsymbol{x}_n-\boldsymbol{\mu}_k)^t}{\sum_{n=1}^{N} q_{nk}} \end{aligned}$$

$$q_{nk} = \frac{\pi_k p\left(\boldsymbol{x}_n|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k\right)}{\sum_{j=1}^{K}\pi_j p\left(\boldsymbol{x}_n|\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j\right)}$$

Compare with maximum likelihood estimation of parameters of a single Gaussian and with posterior probabilities we studied in Bayesian classification.

Expectation Maximization

Auxilliary Variable as Posteriors

Interpret:

- Mixture model as a Gaussian classifier with K classes
- π_k as prior probabilities
- Each of the $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ as class conditional densities / likelihoods.

$$\begin{aligned} p(z_{nk} = 1 | \boldsymbol{x}_n, \boldsymbol{\pi}, \Delta) &= \frac{p(z_{nk} = 1|\pi_k) p(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} p(z_{nk} = 1|\pi_k) p(\boldsymbol{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \\ &= q_{nk} \end{aligned}$$

- Each data item has a weighted contribution to the estimation of parameters.
- Unknown assignment z_{nk}; **E**xpected value of this unknown assignment is q_{nk}, the posterior probability
- **M**aximize (the lower bound) to re-estimate parameters.

K-Means Clustering Algorithm

Input: $\boldsymbol{X} = \{ \boldsymbol{x}_n^t \}_{n=1}^{N}$, K
Output: $\boldsymbol{C}$, `Idx`

`initialize:` $\boldsymbol{C} = \left\{ \boldsymbol{c}_j^t \right\}_{j=1}^{K}$

`repeat`

. assign n^{th} sample to nearest $\boldsymbol{c}_j$

. `Idx(n)` $= \min_j \; ||\boldsymbol{x}_n - \boldsymbol{c}_j||^2$

. recompute $\boldsymbol{c}_j = \frac{1}{N_j} \sum_{n=j} \boldsymbol{x}_n$

`until` no change in $\boldsymbol{c}_1, \; \boldsymbol{c}_2, \; \ldots \; \boldsymbol{c}_k$

`return` $\boldsymbol{C}$, `Idx`

K-Means as Mixture Gaussian

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

- Set $\boldsymbol{\Sigma}_k = \sigma_k^2 \boldsymbol{I}$
- At every iteration, set largest q_{nk} (largest over k) to one and others to zero. Winner take all at each datapoint.
- Computation of q_{nk} is expectation of latent variable z_{nk} – **E** step
- Re-estimation of $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ become maximum likelihood estimates from data assigned to each cluster (because q_{nk} is either one or zero) – **M** step