Expectation Maximization for mixture models

Seong-Wook Joo

1. Motivation

Suppose measurement data $\boldsymbol{X} = \{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N\}$ is i.i.d. from a distribution parameterized by $\boldsymbol{\Theta}$, i.e., $p(\boldsymbol{x}|\boldsymbol{\Theta})$. Then $\boldsymbol{\Theta}$ can be estimated by Maximum-Likelihood:

$$\boldsymbol{\Theta}^* = \text{argmax}_{\boldsymbol{\Theta}} \left[p(\boldsymbol{X}|\boldsymbol{\Theta}) = \Pi_{i=1:N}\, p(\boldsymbol{x}_i|\boldsymbol{\Theta}) \right]$$

But for may problems, it is not possible to find an analytical expression (e.g., mixture model ?)

2. Abstract form of EM

EM can be applied to mainly two types of *missing data problems*: (1) some terms in a data vector are missing (2) (the analytically intractable) likelihood function can be simplified by assuming some missing variables (e.g., classification labels). We concentrate on the latter.

A basic intuition is this: Guess an initial $\boldsymbol{\Theta}$. (E) Using current $\boldsymbol{\Theta}$**,** obtain an expectation of the complete data likelihood. (M) Find (and update) $\boldsymbol{\Theta}^+$ that maximizes the expectation. Here $\boldsymbol{\Theta}$ from the (E) step is a constant. (Fix parameter and get the missing data (and expectation wrt missing data) – fix missing data and find the best parameter - …). Note that “missing data” means the “distribution” of the missing data determined (parameterized) by $\boldsymbol{\Theta}$**.**

Let $\boldsymbol{X}$ be the observed data, $\boldsymbol{Y}$ be the missing data, $\boldsymbol{Z} = (\boldsymbol{X}, \boldsymbol{Y})$ be the complete data. The complete-data likelihood function is defined as

$$L(\boldsymbol{\Theta}|\boldsymbol{Z}) = L(\boldsymbol{\Theta}|\boldsymbol{X}, \boldsymbol{Y}) = p(\boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{\Theta})$$

Initialization: Guess a value of $\boldsymbol{\Theta}$**.**

E-Step:
Find the expected value of the complete-data likelihood with respect to missing $\boldsymbol{Y}$, given observed $\boldsymbol{X}$ and current parameter $\boldsymbol{\Theta}^{i-1}$.

$$Q(\boldsymbol{\Theta},\boldsymbol{\Theta}^{i-1}) = (\text{E}[L(\boldsymbol{\Theta}|\boldsymbol{X}, \boldsymbol{Y})|\, \boldsymbol{\Theta}^{i-1}]) = \text{E}\,[p(\boldsymbol{X}, \boldsymbol{Y}|\boldsymbol{\Theta}) \mid \boldsymbol{X}, \boldsymbol{\Theta}^{i-1}] = \int_{\boldsymbol{y}} p(\boldsymbol{X}, \boldsymbol{y}|\boldsymbol{\Theta})\, \text{f}(\boldsymbol{y} \mid \boldsymbol{X},\boldsymbol{\Theta}^{i-1})\, d\boldsymbol{y}$$

Note that $\text{f}(\boldsymbol{y} \mid \boldsymbol{X},\boldsymbol{\Theta}^{i-1})$ is the distribution of the missing data and is dependent on the observed $\boldsymbol{X}$ and current parameter $\boldsymbol{\Theta}^{i-1}$.

M-Step:
Maximize the expectation from the E-Step wrt $\boldsymbol{\Theta}$. i.e.,

$$\boldsymbol{\Theta} = \text{argmax}_{\boldsymbol{\Theta}}\, Q(\boldsymbol{\Theta},\boldsymbol{\Theta}^{i-1})$$

Repeat the two steps until convergence ($Q(.)$ changes little). The algorithm is guaranteed to converge to a *local* maximum.

3. EM for mixture model – image segmentation

Assume each pixel of an image is associated with a *d*-dimensional feature vector $\boldsymbol{y}$ (e.g., the *d* features include color, texture, etc.) We model our image as a mixture of *g* image segments each of which has its density and weight. That is, in terms of how to produce the image ("generative model"), each pixel vector in the image is obtained by selecting the *l*th segment with probability (mixing weight) π_l and then drawing a sample from the *l*th density:

$$p(\boldsymbol{x}) = \sum \pi_l \, p(\boldsymbol{y}|\theta_l)$$

We would like to determine (a) the parameters of each density (b) the mixing weights (c) the segment label (the missing data). Assuming each component density $p(\boldsymbol{y}|\theta_l)$ is a *d*-dimensional Gaussian with parameter $\theta_l=(\mu_l,\Sigma_l)$, we encapsulate all the parameters into a vector $\boldsymbol{\Theta} = (\pi_1,\ldots,\pi_g, \theta_1,\ldots,\theta_g)$. The log-likelihood for the entire image of *N* pixels is (iid pixels assumed)

$$\sum_{j=1:N} log(\sum_{l=1:g} \pi_l \, p(\boldsymbol{y}|\theta_l))$$

(As can be seen, the $\sum$ inside the log makes it hard to optimize this analytically.) If we knew the missing data, it would be easy to estimate the parameters (via ML) and the mixing weights. If we knew the parameters, we could determine the missing data from the densities. The difficulty is that we know neither the missing data nor the parameters.

To make things easy, lets make the missing variables linear wrt the complete data-log likelihood by the following. Let $\boldsymbol{x}$ be a vector of *g* components. If the label of the corresponding pixel is *l*, then the *l*th component is 1 and the rest are zeros. Then the complete data-log likelihood is

$$\sum_{j=1:N} (\sum_{l=1:g} \boldsymbol{x}_{lj} \, log(\pi_l \, p(\boldsymbol{y}_j \, |\theta_l)))$$

where $\boldsymbol{x}_{lj}$ is the *l*th component of the *j*th vector. Note the inner $\sum$ is simply selecting one of *g* segments' log-likelihood. (Since the likelihood function is linear in $\boldsymbol{x}_{lj}$ we need only take the expectation of $\boldsymbol{x}_{lj}$ instead of the whole function.)

E-Step:

$$I_{lj} \equiv \mathrm{E}(\boldsymbol{x}_{lj}) = \mathrm{P}(j\text{th pixel is from } l\text{th component}) = \pi_l \, p(\boldsymbol{y}_j \, |\theta_l^{(s)}) \, / \, \sum_{k=1:g} \pi_k \, p(\boldsymbol{y}_j \, |\theta_k^{(s)})$$

M-Step:

$$\pi_l^{(s+1)} = 1/N \sum_{j=1:N} I_{lj}$$

$$\mu_l^{(s+1)} = \sum_{j=1:N} \boldsymbol{x}_j \, I_{lj} \, / \, (\sum_{j=1:N} I_{lj})$$

$$\Sigma_l^{(s+1)} = \sum_{j=1:N} \{(\boldsymbol{x}_j-\mu_l^{(s)})(\boldsymbol{x}_j-\mu_l^{(s)})^{\mathrm{T}}\} \, I_{lj} \, / \, (\sum_{j=1:N} I_{lj})$$

Note $\theta_l^{(s+1)}=(\mu_l^{(s+1)}, \Sigma_l^{(s+1)})$ is the weighted mean and covariance.

References

[1] Bilmes, J., "A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models" Technical Report ICSI-TR-97-021, University of Berkeley (1998)

[2] Forsyth and Ponce, "Computer Vision: A Modern Approach" Prentice Hall, 2003, Chapter 16.