67577 – Intro. to Machine Learning **Fall semester, 2004/5**

Lecture 14: EM-II: Soft Clustering, K-means, Gaussian Mixture

Lecturer: Amnon Shashua *Scribe: Amnon Shashua* [1]

In previous lecture we saw that if $\mathbf{x}_i$ is an observation vector (of random variables) and θ is a vector of unknown parameters, then the likelihood $P(\mathbf{x}_i \mid \theta)$ can be represented as a marginal over hidden variables:

$$P(\mathbf{x}_i \mid \theta) = P(\mathbf{x}_i, y_i = 1 \mid \theta) + ... + P(\mathbf{x}_i, y_i = k \mid \theta) = \sum_{j=1}^{k} P(\mathbf{x}_i \mid y_i = j, \theta) P(y_i = j \mid \theta).$$

In the coin example, $k = 2$ and $P(\mathbf{x}_i \mid y_i = 1, \theta) = p^{n_i}(1-p)^{3-n_i}$ a Bernoulli distribution, and $P(y_1 = 1 \mid \theta) = \lambda$ is the probability of $\mathbf{x}_i$ arising from the first factor (coin) in the expansion of $P(\mathbf{x}_i \mid \theta)$. The result is that the likelihood is exapanded as a *mixture* of Bernoulli distributions. In general, we can take any other distribution standing in for the factors in the expansion — say for example a Gaussian distribution.

We saw that EM introduces a new set of variables μ_{ij}:

$$\mu_{ij} = P(y_i = j \mid \mathbf{x}_i, \theta), \qquad \sum_{j=1}^{k} \mu_{ij} = 1$$

which is the probability that $\mathbf{x}_i$ was generated by the j'th factor in the expansion. In the coin example, we had μ_i standing for the probability that $\mathbf{x}_i$ was generated by tossing coin 1 and $1 - \mu_i$ the probability that coin 2 generated the observation.

EM solves for both μ_{ij} and θ by interleaving between the two sets. For data $D = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ i.i.d. we have that:

$$\theta^{(t+1)} = \text{argmax}_\theta \sum_{i=1}^{n} \sum_{j=1}^{k} \mu_{ij}^{(t)} \log \left[P(\mathbf{x}_i \mid y_i = j, \theta) P(y_i = j \mid \theta) \right],$$

where $\mu_{ij}^{(t)}$ were fixed from the previous iteration using the estimate $\theta = \theta^{(t)}$:

$$\mu_{ij}^{(t)} = P(y_i = j \mid \mathbf{x}_i, \theta^{(t)}) \cong P(\mathbf{x}_i \mid y_i = j, \theta^{(t)}) P(y_i = j \mid \theta^{(t)}),$$

where the equality is up to scale, i.e., the μ_{ij} are later scaled such that $\sum_j \mu_{ij} = 1$. We will consider in the next section the case where the factors are Gaussian distributions, thus the likelihood is a mixture of Gaussians.

14.1 Gaussian Mixture

Given $\mathbf{x}_i \in R^d$ and that the factors $P(\mathbf{x}_i \mid y_i = j, \theta)$ are Normally distributed with mean $\mathbf{c}_j$ and variance σ_j^2:

$$P(\mathbf{x}_i \mid y_i = j, \theta) = P(\mathbf{x}_i \mid \theta_j) = \frac{1}{(2\pi)^{d/2}\sigma_j^d} \exp^{-\frac{\|\mathbf{x}_i - \mathbf{c}_j\|^2}{2\sigma_j^2}},$$

[1]

where $\theta_j = (\mathbf{c}_j, \sigma_j)$. Let $\alpha_j = P(y_i = j \mid \theta) = P(\theta_j)$ be the prior of the j'th factor (note that $\sum_j \alpha_j = 1$), then the EM step is:

$$\theta^{(t+1)} = \text{argmax}_\theta \sum_{i=1}^{n}\sum_{j=1}^{k} \mu_{ij}^{(t)} \log\left[P(\mathbf{x}_i \mid \theta_j)\alpha_j\right], \qquad \text{subject to } \sum_{j=1}^{k} \alpha_j = 1. \tag{14.1}$$

The parameters vector θ contains $\theta = (\alpha_j, \mathbf{c}_j, \sigma_j)$, $j = 1, ..., k$. The update formula for μ_{ij} is:

$$\mu_{ij}^{(t)} \cong \alpha_j^{(t)} P(\mathbf{x}_i \mid \theta^{(t)}),$$

where the equality is up to scale, i.e., the μ_{ij} are later scaled such that $\sum_j \mu_{ij} = 1$. The update formula for $\alpha_j, \mathbf{c}_j, \sigma_j$ follow by taking partial derivatives of the Lagrangian of eqn. (14.1). The Lagrangian $L(\theta, \lambda)$ is:

$$L(\theta, \lambda) = \sum_{i=1}^{n}\sum_{j=1}^{k} \mu_{ij} \log \alpha_j + \sum_{i=1}^{n}\sum_{j=1}^{k} \mu_{ij} \log P(\mathbf{x}_i \mid \theta_j) + \lambda(\sum_j \alpha_j - 1),$$

where λ is the Lagrange multiplier due to the equality constraint. The partial derivative with respect to α_j is:

$$\frac{\partial L}{\partial \alpha_j} = \frac{1}{\alpha_j}\sum_i \mu_{ij} + \lambda = 0.$$

Since this holds for all $j = 1, ..., k$ we can recover the value of λ by summing the constraint above over j , i.e.,

$$\sum_j (\sum_i \mu_{ij} + \alpha_j \lambda) = \sum_i \sum_j \mu_{ij} + \lambda \sum_j \alpha_j,$$

from which we obtain $\lambda = -n$ and as a result:

$$\alpha_j = \frac{1}{n}\sum_{i=1}^{n} \mu_{ij}.$$

In other words, the updated prior $P(\theta_j)$ is the average of $P(y_i = j \mid \mathbf{x}_i, \theta)$ over the observations. Taking partial derivatives with respect to $\mathbf{c}_j$ and σ_j we obtain the update rules:

$$\begin{aligned} \mathbf{c}_j &= \frac{1}{\sum_i \mu_{ij}} \sum_{i=1}^{n} \mu_{ij}\mathbf{x}_i, \\ \sigma_j^2 &= \frac{1}{d\sum_i \mu_{ij}} \sum_{i=1}^{n} \mu_{ij} \|\mathbf{x}_= \mathbf{c}_j\|^2. \end{aligned}$$

In other words, the observations $\mathbf{x}_i$ are weighted by μ_{ij} before a Gaussian is fitted (k times, one for each factor).

14.2 EM and K-means

In previous lecture and now with the derivation of the Gaussian mixture model, we see that the $\mu_{ij} = P(y_i = j \mid \mathbf{x}_i, \theta)$ play the role of a "probabilistic factor assignment". What the EM scheme does is that it turns the ML problem into a probabilistic clustering problem of assigning the

indices $\{1, ..., n\}$ into k clusters. At each EM iteration, the probabilistic assignments μ_{ij} are re-estimated (given the current estimate $\theta^{(t)}$) in order to provide the most updated estimation as to the probability of $\mathbf{x}_i$ to arise from each of the k factors.

This assignment can be considered as a "soft" clustering. A "hard" decision can be made by associating $\mathbf{x}_i$ with cluster j if $\mu_{ij} > \mu_{ir}$, $r \neq j$. If a new observation $\mathbf{x}$ is introduced after the EM has converged and outputed the parameters θ_j, the cluster assignment of $\mathbf{x}$ can be made by computing $\mu_j = P(y = j \mid \mathbf{x}, \theta)$ using the Bayes rule.

We could turn this around; say we are interested in clustering $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ points in R^d where we assume that the clusters are Normally distributed (a reasonable assumption in many practical settings). Then the EM algorithm is exactly the clustering method that would solve for the cluster assignments by recovering the underlying Normal distributions.

In this regard, the EM algorithm can be considered as an extension of a "hard" clustering method known as "K-means" where the assumption is that the clusters are defined by their mean vectors only. In other words, the goal of clustering is to find those k mean vectors $\mathbf{c}_1, ..., \mathbf{c}_k$ and provide the cluster assignment of each point in the set. The K-means algorithm is also based an interleaving approach where the cluster assignments are established given the centers and the centers are computed given the assignments. The optimization criteria is as follows:

$$\min_{y_1,...,y_n,\mathbf{c}_1,...,\mathbf{c}_k} \sum_{j=1}^{k} \sum_{y_i=j} \|\mathbf{x}_i - \mathbf{c}_j\|^2$$

Assume that $\mathbf{c}_1, ..., \mathbf{c}_k$ are given from the previous iteration, then

$$y_i = \text{argmin}_j \|\mathbf{x}_i - \mathbf{c}_j\|^2,$$

and next assume that $y_1, .., y_n$ (cluster assignments) are given, then for any set $S \subseteq \{1, ..., n\}$ we have that

$$\frac{1}{|S|} \sum_{j \in S} \mathbf{x}_j = \text{argmin}_{\mathbf{c}} \sum_{j \in S} \|\mathbf{x}_j - \mathbf{c}\|^2.$$

In other words, given the estimated centers in the current round, the new assignments are computed by the closest center to each point $\mathbf{x}_i$, and then given the updated assignments the new centers are estimated by taking the mean of each cluster.

We see that K-means makes a "hard" assignment in each iteration whereas EM makes a probabilistic assignment. The drawback of a hard assignment is that a small shift of a data point can flip it to a different cluster. Moreover, the EM allows for a more complex cluster "shapes" (such as Gaussians) than K-means which relies only on the means.