STATS 306B: Unsupervised Learning **Spring 2014**

Lecture 2 — April 2

Lecturer: Lester Mackey *Scribe: Junyang Qian, Minzhe Wang*

2.1 Recap

In the last lecture, we formulated our working definition of **unsupervised learning**: discovering the latent structure underlying our data, without prior observations of that structure. Our exploration of the field by began with the task of **clustering**, in which we aim to group or segment data points based on some notion of similarity. In particular, we studied the k**-means** approach to clustering and motivated it entirely in terms of minimizing an objective

$$\sum_{i=1}^{n} ||x_i - m_{z_i}||_2^2$$

that characterizes the similarity of data points to their cluster means. This is an example of a model-free approach to clustering, as it makes no explicit attempt to model the process that generated our observations. In this lecture, we will examine a popular alternative to k-means clustering – Gaussian mixture modeling with Expectation-Maximization – that reposes upon an explicit model of the data-generating process.

2.2 Gaussian Mixture Modeling

The **Gaussian mixture model** (GMM) is a probabilistic model for clustered data with real-valued components. Although the aims and assumptions of Gaussian mixture modeling appear to be quite different from those of k-means, we will see soon that they share some key similarities.

2.2.1 Model formulation

We begin by positing a **statistical model** for our data. That is, we view our data $X_1, X_2, \cdots, X_n$ as random variables drawn *i.i.d.* from an *unknown* distribution with density $p(x)$. A GMM demands a specific form for this density,

$$p(x) = \sum_{j=1}^{k} \pi_j \phi(x; \mu_j, \Sigma_j).$$

This is a mixture of k component multivariate Gaussian distributions where

- $\phi(x; \mu_j, \Sigma_j) = \frac{1}{|2\pi\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j)\right)$ is a multivariate Gaussian density with unknown parameters (μ_j, Σ_j), and

- π_j is the unknown probability of selecting component j, satisfying $\sum_{j=1}^{k} \pi_j = 1$.

A GMM has an equivalent representation as a **generative model** for our data:

$$\begin{aligned} z_i &\overset{\text{iid}}{\sim} \text{Mult}(\pi, 1) \\ x_i | z_i &\overset{\text{ind}}{\sim} \mathcal{N}(\mu_{z_i}, \Sigma_{z_i}). \end{aligned}$$

Here, z_i represents the latent component indicator or latent class / cluster for datapoint x_i.

Remark. Throughout the course, we will be considering a number of probabilistic modeling approaches to unsupervised learning. Typically, we will not view these models as literal descriptions of reality but rather as convenient modeling frameworks that give rise to procedures for unsupervised learning.

2.2.2 Clustering with GMMs

Under the GMM, our clustering task amounts to inferring the latent component z_i responsible for each x_i. For the moment, we will ignore the fact that we do not know the parameters of the GMM and imagine how we would carry out the clustering task given $(\pi, \mu_{1:k}, \Sigma_{1:k})$. Since a GMM with known parameters defines a joint distribution over (x_i, z_i), it is natural to consider the conditional distribution of each z_i given x_i:

$$\begin{aligned} p(z_i = j | x_i) &= \frac{p(z_i = j) p(x_i | z_i = j)}{p(x_i)} \\ &= \frac{\pi_j \phi(x_i; \mu_j, \Sigma_j)}{\sum_{l=1}^{k} \pi_l \phi(x_i; \mu_l, \Sigma_l)}. \end{aligned}$$

These conditionals reflect our updated beliefs concerning z_i after x_i is observed: before we observe x_i, we have the prior belief that it belongs to cluster j with probability π_j; after observing x_i, we can update this belief in accordance with the likelihood of x_i under each Gaussian component.

Remark. The task of computing conditionals or marginals from a known joint distribution is sometimes called **probabilistic inference**.

The conditional distribution provides us with what is called a **soft clustering** since it assigns some probability to x_i belonging to each cluster. To obtain a **hard clustering** (an assignment of x_i to a single cluster), one typically selects a mode of the conditional distribution $\text{argmax}_j \, p(z_i = j | x_i)$.

Remark. Those familiar with discriminant analysis will notice a relationship with the classification rule in quadratic and linear discriminant analysis. This is to be expected as discriminant analysis also models datapoints as being drawn from class conditional multivariate Gaussian distributions. However, LDA operates in a supervised setting, which greatly simplifies the task of parameter estimation. Parameter estimation is a good deal more difficult in our unsupervised GMM setting.

2.2.3 Estimating GMM parameters with Expectation-Maximization

In the prior section, we carried out clustering assuming that the GMM parameters $(\pi, \mu_{1:k}, \Sigma_{1:k})$ were known. In this section, we will attempt to estimate these parameters, a task sometimes termed **statistical inference** to distinguish it from probabilistic inference. We will begin by trying to derive maximum likelihood estimates (MLEs) of the GMM parameters.

Consider the log likelihood of our data

$$\sum_{i=1}^{n} \log(p(x_i)) = \sum_{i=1}^{n} \log(\sum_{j=1}^{k} \pi_j \phi(x_i; \mu_j, \Sigma_j)). \tag{2.1}$$

The **maximum likelihood principle** would choose estimates of $(\pi, \mu_{1:k}, \Sigma_{1:k})$ that maximize this expression. When $k = 1$ (i.e., when there is no clustering structure), the log likelihood simplifies to

$$\sum_{i=1}^{n} \log(\phi(x_i; \mu_j, \Sigma_j)) = \sum_{i=1}^{n} \left[-\frac{1}{2}(x_i - \mu_1)^T \Sigma_1^{-1} (x_i - \mu_1) - \log |2\pi\Sigma_1|^{1/2} \right],$$

which has closed-form maxima,

$$(\mu_1^*, \Sigma_1^*) = (\frac{1}{n} \sum_{i=1}^{n} x_i, \frac{1}{n} \sum_{1=1}^{n} (x_i - \mu_1^*)(x_i - \mu_1^*)^T).$$

Unfortunately, when $k > 1$ (the case of interest), the log likelihood no longer simplifies and no longer yields closed-form solutions. To find (approximate) MLEs one often turns to (I) off-the-shelf numerical optimizers or (II) the Expectation-Maximization (EM) algorithm, which leverages the latent-variable problem structure to form parameter estimates. We will develop and examine the EM approach in the remainder of this lecture.

The fundamental difficulty with the log likelihood in (2.1) is the sum inside of each log (representing an expectation over an unknown cluster assignment z_i), which couples all of the parameters of all of the component Gaussian distributions of the mixture together. If we knew the z_i's, then this problem would be solved, since we could instead maximize the **complete log likelihood**,

$$\begin{aligned}
\sum_{i=1}^{n} \log(p(x_i, z_i)) &= \sum_{i=1}^{n} (\log(p(x_i|z_i)) + \log(p(z_i))) \\
&= \sum_{i=1}^{n} (\log(\phi(x_i; \mu_{z_i}, \Sigma_{z_i})) + \log(\pi_{z_i})) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{k} (\mathbb{I}[z_i = j] \log \phi(x_i; \mu_j, \Sigma_j) + \mathbb{I}[z_i = j] \log \pi_j).
\end{aligned}$$

Note that the sum over z_i values is now outside of the log and that parameters of different Gaussian components no only appear in separate summands.

Moreover, the complete log likelihood, viewed as a function of $(\pi, \mu_{1:k}, \Sigma_{1:k})$, has closed-form maxima:

$$\pi_j^* = \frac{1}{n}\sum_{i=1}^n \mathbb{I}[z_i = j],$$

$$\mu_j^* = \frac{\sum_{i=1}^n \mathbb{I}[z_i = j]x_i}{\sum_{i=1}^n \mathbb{I}[z_i = j]},$$

and

$$\Sigma_j^* = \frac{\sum_{i=1}^n \mathbb{I}[z_i = j](x_i - \mu_j^*)(x_i - \mu_j^*)^T}{\sum_{i=1}^n \mathbb{I}[z_i = j]}.$$

That is, π_j^* equals to the proportion of sample points assigned to cluster j, and μ_j^* and Σ_j^* are the sample mean and covariance of points within cluster j. Hence, if we knew the cluster assignments ahead of time, we could easily estimate the GMM parameters, and, as mentioned previously, if we knew the parameters, we could easily estimate cluster assignments using probabilistic inference. Unfortunately (as is often the case in unsupervised learning), we know neither, so we will simply guess the values of the GMM and iteratively refine our guess by alternating between probabilistic inference (updating our cluster assignment inferences) and statistical inference (updating our parameter estimates). This is the Expectation-Maximization algorithm for parameter estimation in a nutshell. More precisely, the algorithm consists of the following steps.

Expectation-Maximization for GMMs:

0. Initialize $\pi, (\mu_{1:k}, \Sigma_{1:k})$ arbitrarily

1. Alternate until convergence

   (E-step) [Expectation step]: Compute soft class memberships, given the current parameters:

   $$\tau_{ij} = P(z_i = j | x_{ij}, \pi, (\mu_\ell, \Sigma_\ell)).$$

   (M-step) [Maximization step]: Update parameters by plugging in τ_{ij} (our guess) for the unknown $\mathbb{I}[z_i = j]$, which gives us:

   $$\pi_j = \frac{1}{n}\sum_{i=1}^n \tau_{ij}, \quad \mu_j = \frac{\sum_{i=1}^n \tau_{ij} x_i}{\sum_{i=1}^n \tau_{ij}},$$

   $$\Sigma_j = \frac{\sum_{i=1}^n \tau_{ij}(x_i - \mu_j)(x_i - \mu_j)^T}{\sum_{i=1}^n \tau_{ij}}.$$

   This is similar to the case in which all z_i's are known, but now each x_i is partially assigned to each cluster j through the conditional probability that $z_i = j$.

Note the similarity to Lloyd's algorithm for k-means discussed last time. If we ignore the updates to π and $\Sigma_{1:k}$, we see the same basic structure: assign data points to classes, and update the means according to the assigned classes. There are also important distinctions between the two algorithms:

- Lloyd's algorithm makes hard assignments on each iteration, with each point assigned to one class, while the EM algorithm uses soft, probabilistic assignments, where conditional probabilities are computed for each class given the parameter estimates.
- In the GMM setting, we model the mixture proportions and covariance structure in the data. This is especially useful if we have correlated features, features of with varying variances, or clusters of varying sizes. Meanwhile, k-means effectively assumes identity covariance structure (spherical clusters) and equal cluster sizes.

Interestingly, we can actually recover the Lloyd's algorithm from a variant of the GMM EM algorithm above. If we assume the cluster proportions $\pi_j = \frac{1}{k}$, and we also assume $\Sigma_j = \sigma^2 I$ for some known σ^2, then the EM algorithm (with π and $\Sigma_{1:k}$ fixed and known) updates the means μ_j and becomes Lloyd's algorithm as $\sigma^2 \to 0$. The design of fast unsupervised learning algorithms like k-means from probabilistic models using **small variance asymptotics** is currently an active research area.

At this point, one should have a great many questions concerning this mysterious EM algorithm. For example,

- Does it converge? If so, are rates of convergence known?
- Are its solutions optimal? If not, can we bound their suboptimality?
- What is its relationship to the likelihood and our goal of maximizing it?

To provide satisfying answers to these questions, we will, in the next lecture, turn to a more general view of EM for estimation in *any* latent variable model. In the remainder of this lecture, we will content ourselves with a teaser indicating a fundamental relationship between EM and maximum likelihood.

2.2.4 Optimality conditions for maximum likelihood

To understand how EM might relate to MLE, let us consider the first-order optimality conditions for maximum likelihood. We begin by computing the partial derivatives of the log-likelihood with respect to μ_j,

$$\begin{aligned}
&\frac{\partial}{\partial \mu_j}\left\{\sum_{i=1}^{n} \log\Big[\sum_{i=1}^{k} \pi_j \phi(x_i;\mu_j,\Sigma_j)\Big]\right\} \\
=\;& \sum_{i=1}^{n} \frac{\pi_j \phi(x_i;\mu_j,\Sigma_j)}{\sum_\ell \pi_\ell \phi(x_i;\mu_\ell,\Sigma_\ell)} \cdot \frac{\partial}{\partial \mu_j} \log \phi(x_i;\mu_j,\Sigma_j) \\
=\;& \sum_{i=1}^{n} \tau_{ij} \Sigma_j^{-1}(x_i - \mu_j),
\end{aligned}$$

where the second line follows from the chain rule, and, in the third line, we have recognized the expressions for our previously-computed conditional probabilities τ_{ij}. The first-order condition for optimality requires that these derivatives equal zero. Hence, any maximum likelihood solution $(\pi^*, \mu^*_{1:k}, \Sigma^*_{1:k})$ satisfies

$$\sum_{i=1}^{n} \tau_{ij} \Sigma_j^{*-1}(x_i - \mu_j^*) = 0 \Rightarrow \mu_j^* = \frac{\sum_{i=1}^{n} \tau_{ij} x_i}{\sum_{i=1}^{n} \tau_{ij}}.$$

This derived requirement has the same form as the M-step update from EM! However, the τ_{ij} on the RHS of this expression depends implicitly on the optimal parameters $(\pi^*, \mu^*_{1:k}, \Sigma^*_{1:k})$. Hence, this is not a closed-form solution for μ_j^* but rather a fixed-point equation. (Similar fixed-point equations would result from considering the first-order optimality conditions for π and Σ.) This derivation makes clear that EM performs **fixed-point iteration** on the optimality equations for likelihood maximization; that is, EM iteratively plugs in estimates of τ_{ij} into the RHS of the fixed-point equations based on current parameter estimates and uses the results as its updated parameter estimates. We will explore additional properties of the EM algorithm next time.