National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”
Institute of Physics and Technology

Lecture 9
Generative Models for Discrete Data. Mixture Models

Dmytro Progonov,
PhD, Associate Professor,
Department Of Physics and Information Security Systems

Content

- Latent variable models;
- Mixture model;
- Parameters estimation for mixture model;
- The EM algorithm;
- EM for GMM;
- Other EM variants.

Latent variable models

One of common approach to define high-dimensional joint-probability distribution is usage of Graphical Models – to model dependence between two variables by adding an edge between them in graph.

Alternative approach is to assume that observed variables are correlated because they arise from a hidden common "cause" – ***latent variable models*** (***LVM***). *Advantages of LVMs are*:

1. LVMs often have fewer parameters than models that directly represent correlation in the visible space;
2. Hidden variables in an LVM can serve as a bottleneck, which computes a compressed representation of the data.

Mixture model

The simplest form of LVM is when $z_i \in \{1,2,3 \cdots K\}$, representing a discrete latent state. The overall model known as a ***mixture model***, since we are mixing together the K base distributions p_k:

$$p(\mathbf{x}_i|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{x}_i|\boldsymbol{\theta})$$

Where π_k is ***mixing weights*** satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$

Examples of mixture model

Mixtures of Gaussians (***Gaussian Mixture Model***):

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

Mixtures of Multinoullis:

$$p(\mathbf{x}_i|z_i = k, \boldsymbol{\theta}) = \prod_{j=1}^{D} Ber(x_{ij}|\mu_{jk}) = \prod_{j=1}^{D} {\mu_{jk}}^{x_{ij}} (1 - \mu_{jk})^{1-x_{ij}}$$

Mixtures of experts:

$$\begin{cases} p(y_i|\mathbf{x}_i, z_i = k, \boldsymbol{\theta}) = \mathcal{N}(y_i|{\mathbf{w}_k}^T \mathbf{x}_i, {\sigma_k}^2) \\ p(z_i|\mathbf{x}_i, \boldsymbol{\theta}) = Cat(z_i|\mathcal{S}(\mathbf{V}^T \mathbf{x}_i)) \end{cases}$$

Parameters estimation for mixture model

Since in an LVM some parameters are hidden, the parameters are not longer independent, and the posterior does not factorize, making it much harder to compute MAP and ML:

1. *Unidentifiability* – there is not a unique MLE;
2. *Computing a MAP estimate is non-convex* – requires using multiple random restart in practice to increase our chance of finding a "good" local optimum.

The EM algorithm (1/3)

Common approach to find a local minimum of the ***negative log likelihood*** is to use a generic gradient-based optimizer:

$$NLL(\boldsymbol{\theta}) \triangleq -\frac{1}{N}\log[p(\mathcal{D}|\boldsymbol{\theta})]$$

However, we often have to enforce constrains, such as the fact that covariance matrices must be positive defined, mixing weights must sum to one etc., which can be tricky. In such cases, it is often much simpler (but not always faster) to use the ***expectation maximization*** (***EM***) algorithm.

EM exploits the fact that if the data were fully observed, then the ML/MAP estimate would be easy to compute. In particular, EM is an iterative algorithm which alternates between *inferring the missing values* given the parameters (*E step*), and then *optimizing the parameters* given the "filled in" data (*M step*).

The EM algorithm (2/3)

Let $\mathbf{x}_i$ be the visible or observed variables in case i, and let $\mathbf{z}_i$ be the hidden or missing variables. The goal is to maximize the log likelihood of the observed data:

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log[p(\mathbf{x}_i|\boldsymbol{\theta})] = \sum_{i=1}^{N} \log\left[\sum_{\mathbf{z}_i} p(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\theta})\right]$$

Let us define the ***complete data log likelihood*** to be:

$$\ell_C(\boldsymbol{\theta}) \triangleq \sum_{i=1}^{N} \log[p(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\theta})]$$

The EM algorithm (3/3)

This cannot be computed since $\mathbf{z}_i$ is unknown. So let us define the ***expected complete data log likelihood*** as follows

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}) = \mathbb{E}[\ell_C(\boldsymbol{\theta}) | \mathcal{D}, \boldsymbol{\theta}^{t-1}]$$

where t is the current iteration number. Q is called the ***auxiliary function***. The expectation is taken wrt the old parameters $\boldsymbol{\theta}^{t-1}$, and the observed data $\mathcal{D}$.

The goal of E step is to compute $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1})$, or rather, the term inside of it which the MLE depends on. These are known as the ***expected sufficient statistics*** or ***ESS***.

In the M step we optimize the Q function wrt $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^t = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1})$$

EM for GMM (1/3)

Let us discuss how to fit a mixture of Gaussian (GMM) using EM. WE assume the number of mixture components K is known.

Auxiliary function

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}) \triangleq \mathbb{E}\left[\sum_i \log[p(\mathbf{x}_i, \mathbf{z}_i|\boldsymbol{\theta})]\right] = \sum_i \mathbb{E}\left[\log\left[\prod_{k=1}^{K} (\pi_k p(\mathbf{x}_i|\boldsymbol{\theta}_k))^{\mathbb{I}(z_i=k)}\right]\right]$$

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{t-1}) = \sum_i \sum_k r_{ik} \log[\pi_k] + \sum_i \sum_k r_{ik} \log[p(\mathbf{x}_i|\boldsymbol{\theta}_k)]$$

where $r_{ik} \triangleq p(z_i = k|\mathbf{x}_i, \boldsymbol{\theta}^{t-1})$ is the responsibility that cluster k takes for data point i. This is computed in the E step.

EM for GMM (2/3)

The *E step* has the following simple form, which is the same for any mixture model:

$$r_{ik} = \frac{\pi_k p\left(\mathbf{x}_i, \boldsymbol{\theta}_k{}^{t-1}\right)}{\sum_{\acute{k}} \pi_{\acute{k}} p\left(\mathbf{x}_i, \boldsymbol{\theta}_{\acute{k}}{}^{t-1}\right)}$$

In the *M step*, we optimize Q wrt $\boldsymbol{\pi}$ and the $\boldsymbol{\theta}_k$. For $\boldsymbol{\pi}$, we obviously have

$$\pi_k = \frac{1}{N}\sum_i r_{ik} = \frac{r_k}{N}$$

To derive the M step for the $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ terms, we look at the parts of Q that depend on $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$. The result is

$$\ell(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \sum_k \sum_i r_{ik} \log[p(\mathbf{x}_i|\boldsymbol{\theta}_k)] = \left(-\frac{1}{2}\right)\sum_i r_{ik}\left[\log[\boldsymbol{\Sigma}_k] + (\mathbf{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k{}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k)\right]$$

EM for GMM (3/3)

This is just a weighted version of the standard problem of computing the MLEs of an MVN. One can show that the new parameter estimates are given by:

$$\boldsymbol{\mu}_k = \frac{\sum_i r_{ik}\mathbf{x}_i}{r_k}$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_i r_{ik}(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T}{r_k} = \frac{\sum_i r_{ik}\mathbf{x}_i\mathbf{x}_i{}^T}{r_k} - \boldsymbol{\mu}_k\boldsymbol{\mu}_k{}^T$$

These equations make intuitive sense: the mean of cluster k is just the weighted average of all points assigned to cluster k, and the covariance is proportional to the weighted empirical scatter matrix.

After computing the new estimates, we set $\boldsymbol{\theta}^t = (\pi_k, \mu_k, \boldsymbol{\Sigma}_k)$ for $k = 1{:}K$, and go to the next E step.

Other EM variants

- Annealed EM;
- Variational EM;
- Monte Carlo EM;
- Generalized EM;
- Expectation Conditional Maximization (ECM) algorithm;
- Over-relaxed EM.

Note that EM in fact just a special case of a larger class of algorithms known as ***bound optimization*** or ***minorize-maximize*** (***MM***) algorithms

Conclusion

- Latent variable and mixture models were considered;
- Procedures for parameters estimation for mixture model such as Expectation Maximization algorithm were presented;
- Generalization and special types of EM algorithm were shown.