# LVMs (Contd), Expectation Maximization (1)

CS771: Introduction to Machine Learning

Piyush Rai

# Plan

- ALT-OPT and EM
  - Example: Gaussian Mixture Model for data clustering
- A deeper look at ALT-OPT and EM
- General recipe for doing ALT-OPT and EM for any LVM

# Need for EM/ALT-OPT: Two Equivalent Perspectives

1. Consider an LVM with latent variables and parameters. Trying to estimate parameters without also estimating the latent variables (by marginalizing them) is difficult

A Gaussian Mixture Model (GMM)

$$p(\boldsymbol{x}_n|\Theta) = \sum_{k=1}^{K} p(\boldsymbol{x}_n, \boldsymbol{z}_n = k|\Theta) = \sum_{k=1}^{K} p(\boldsymbol{z}_n = k|\phi)p(\boldsymbol{x}_n|\boldsymbol{z}_n = k, \theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

This issue not just for MLE for GMM but MLE for other LVMs too

MLE for GMM with cluster ids marginalized/summed/integrated out

$$\Theta_{MLE} = \underset{\Theta}{\operatorname{argmax}} \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

Can't get closed form expressions for the $\pi_k, \mu_k, \Sigma_k$ due to "log of sum". Have to use gradient based methods

EM/ALT-OPT will help us "simulate" this condition by making guesses about the values of $\boldsymbol{z}_n$'s

If we knew the $\boldsymbol{z}_n$'s, the problem will be much simpler; just like MLE for generative classification with Gaussian class-conditional

Since no marginalization of the $\boldsymbol{z}_n$'s required

2. Consider a complex prob. density (without any latent vars) for which MLE is hard

Directly defining a probability density as a mixture of Gaussians ($\boldsymbol{x}_n$ is generated by the $k^{th}$ Gaussian with probability $\pi_k$) without any reference to any latent variable whatsoever (we didn't define it as an LVM)

$$p(\boldsymbol{x}_n|\Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

MLE for the params $\Theta$ of this distribution will again be hard (as we already saw above). However, we can artificially introduce a latent variable $z_n$ with each data point $x_n$, denoting which Gaussian generated $x_n$

Now this prob. density estimation problem also becomes Problem 1 above - a clustering problem with latent variables

Can now apply ALT-OPT/EM to estimate parameters $\Theta$ + we get the latent variables $z_n$ as a "by-product" (though we may not be interested in learning $z_n$'s if our goal is just density estimation, not clustering)

Even though we didn't need the artificially introduced $z_n$'s, their presence and doing ALT-OPT/EM made our job of estimating $\Theta$ easier!

Also in any LVM, given $\Theta$, you can always estimate $z_n$'s. Likewise, given $z_n$, you can always estimate $\Theta$

Remember that GMM is just like generative classification with Gaussian class-conditionals and training data labels unknown

# ALT-OPT/EM for Gaussian Mixture Model

# Detour: MLE for Generative Classification

- Assume a $K$ class generative classification model with Gaussian class-conditionals
- Assume class $k = 1,2,\dots,K$ is modeled by a Gaussian with mean $\mu_k$ and cov matrix $\Sigma_k$
- Can assume label $y_n$ to be one-hot and then $y_{nk} = 1$ if $y_n = k$, and $y_{nk} = 0$, o/w
- Assuming class prior as $p(y_n = k) = \pi_k$, the model has params $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$
- Given training data $\{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$, the MLE solution will be

$$\hat{\pi}_k = \frac{1}{N}\sum_{n=1}^{N} y_{nk}$$

Same as $\frac{N_k}{N}$ where $N_k$ is # of training ex. for which $y_n = k$

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

Same as $\frac{1}{N_k}\sum_{n:y_n=k}^{N} \boldsymbol{x}_n$

$$\hat{\Sigma}_k = \frac{1}{N_k}\sum_{n=1}^{N} y_{nk}(\boldsymbol{x}_n - \hat{\mu}_k)(\boldsymbol{x}_n - \hat{\mu}_k)^\top$$

Same as $\frac{1}{N_k}\sum_{n:y_n=k}^{N} (\boldsymbol{x}_n - \hat{\mu}_k)(\boldsymbol{x}_n - \hat{\mu}_k)^\top$

# Detour: MLE for Generative Classification

- Here is a formal derivation of the MLE solution for $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$

multinoulli

Gaussian

$$\hat{\Theta} = \text{argmax}_\Theta\, p(\boldsymbol{X}, \boldsymbol{y}|\Theta) = \text{argmax}_\Theta \prod_{n=1}^{N} p(\boldsymbol{x}_n, y_n|\Theta)$$

$$= \text{argmax}_\Theta \prod_{n=1}^{N} p(y_n|\Theta)\, p(x_n|y_n, \Theta)$$

$$= \text{argmax}_\Theta \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{y_{nk}} \prod_{k=1}^{K} p(x_n|y_n = k, \Theta)^{y_{nk}}$$

$$= \text{argmax}_\Theta \prod_{n=1}^{N} \prod_{k=1}^{K} [\pi_k p(x_n|y_n = k, \Theta)]^{y_{nk}}$$

$$= \text{argmax}_\Theta \log \prod_{n=1}^{N} \prod_{k=1}^{K} [\pi_k p(x_n|y_n = k, \Theta)]^{y_{nk}}$$

$$= \text{argmax}_\Theta \sum_{n=1}^{N} \sum_{k=1}^{K} y_{nk}[\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)]$$

In general, in models with probability distributions from the exponential family, the MLE problem will usually have a simple analytic form

Also, due to the form of the likelihood (Gaussian) and prior (multinoulli), the MLE problem had a nice separable structure after taking the log

Can see that, when estimating the parameters of the $k^{th}$ Gaussian $(\pi_k, \mu_k, \Sigma_k)$, we only will only need training examples from the $k^{th}$ class, i.e., examples for which $y_{nk} = 1$

The form of this expression is important; will encounter this in GMM too

# Detour: Exponential Family

Exp-fam dist also used for Genealized Linear Models (GLM) with $p(y|\boldsymbol{x},\boldsymbol{w})$ modeled by an exp-fam distribution whose natural parameter is defined by $\boldsymbol{w}^\top\boldsymbol{x}$ (thus "linear"). Useful in problems where $y$ is not real/categorical but a count, or positive real, etc

Lin reg, logistic reg, softmax reg are also instances of GLMs

- Exponential Family is a family of prob. distributions that have the form

$$p(x|\theta) = h(x)\exp[\theta^\top T(x) - A(\theta)]$$

Even though their standard form may not look like this, they can be rewritten in this form after some algebra

- Many well-known distribution (Bernoulli, Binomial, multinoulli, Poisson, beta, gamma, Gaussian, etc.) are examples of exponential family distributions
- $\theta$ is called the natural parameter of the family

Natural params are a function of the distribution parameters in the standard form

- $h(x), T(x)$, and $A(\theta)$ are known functions (specific to the distribution)
- $T(x)$ is called the sufficient statistics: estimates of $\theta$ contain $x$ in form of suff-stats
- Every exp. family distribution also has a conjugate distribution (often also in exp. family)
- Also, MLE/MAP is usually quite simple since $\log p(x|\theta)$ will have a simple expression
- Also useful in fully Bayesian inference since they have conjugate priors

https://en.wikipedia.org/wiki/Exponential_family

# MLE for GMM

- Already saw that MLE is hard for GMM

$$\Theta_{MLE} = \underset{\Theta}{\operatorname{argmax}} \log p(\boldsymbol{X}|\Theta) = \underset{\Theta}{\operatorname{argmax}} \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$$

- Two possible ways to solve this MLE problem

Will soon see how to get these guesses

1. If someone gave us optimal "point" guesses $\hat{\boldsymbol{z}}_n$'s of cluster ids $\boldsymbol{z}_n$'s, we could do MLE for the parameters just like we did for generative classification with Gaussian class-conditionals

$$\Theta_{MLE} = \underset{\Theta}{\operatorname{argmax}} \log p(\boldsymbol{X}, \hat{\boldsymbol{Z}}|\Theta) = \operatorname{argmax}_{\Theta} \sum_{n=1}^{N} \sum_{k=1}^{K} \hat{z}_{nk}[\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)]$$

In form of a probability distribution instead of a singe "optimal" guess

2. Alternatively, if someone gave a "probabilistic" guess of $\boldsymbol{z}_n$'s, we can do MLE for $\boldsymbol{\Theta}$ as follows

$$\Theta_{MLE} = \underset{\Theta}{\operatorname{argmax}} \mathbb{E}[\log p(\boldsymbol{X}, \boldsymbol{Z}|\Theta)] = \operatorname{argmax}_{\Theta} \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}[z_{nk}][\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)]$$

Similar to Approach 1 but maximizes an expectation

The expectation is w.r.t a distribution of Z which we will see shortly

- Approach 1 is **ALT-OPT** and Approach 2 is **Expectation Maximization** ("soft" ALT-OPT). Both require alternating between estimating $\boldsymbol{Z}$ and $\boldsymbol{\Theta}$ until convergence

# ALT-OPT for GMM

Keep in mind: In LVMs, assuming i.i.d. data, the quantity $\log p(\boldsymbol{X}|\Theta) = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\Theta)$ is called incomplete data log-likelihood (ILL) whereas $\log p(\boldsymbol{X}, \boldsymbol{Z}|\Theta) = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta)$ is called complete data log-likelihood (CLL). Goal is to maximize ILL but ALT-OPT maximizes CLL (EM too will maximize the expectation of CLL). The latent vars $\boldsymbol{z}_n$'s "complete" the data $\boldsymbol{x}_n$ ☺

- We will assume we have a "hard" (most probable) guess of $\boldsymbol{z}_n$, say $\hat{\boldsymbol{z}}_n$
- Then ALT-OPT would look like this
  - Initialize $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ as $\widehat{\Theta}$
  - Repeat the following until convergence
    - For each $n$, compute most probable value (our best guess) of $z_n$ as

$$\hat{z}_n = \text{argmax}_{k=1,2,\ldots,K}\, p(z_n = k|\widehat{\Theta}, \boldsymbol{x}_n)$$

Proportional to prior prob times likelihood, i.e., $p(z_n = k|\Theta)\, p(x_n|z_n = k, \Theta) = \pi_k\, \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)$

Posterior probability of point $x_n$ belonging to cluster $k$

    - Solve MLE problem for $\Theta$ using most probable $z_n$'s

$$\widehat{\Theta} = \text{argmax}_{\Theta} \sum_{n=1}^{N} \sum_{k=1}^{K} \hat{z}_{nk}[\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)]$$

Same objective function as generative $K$-class classification with Gaussian class-conditionals

Note: The objective function is $\sum_{n=1}^{N} \log p(\boldsymbol{x}_n, \hat{\boldsymbol{z}}_n|\Theta) = \sum_{n=1}^{N} \log p(\hat{\boldsymbol{z}}_n|\Theta) + \log p(\boldsymbol{x}_n|\hat{\boldsymbol{z}}_n, \Theta)$

$$\hat{\pi}_k = \frac{1}{N} \sum_{n=1}^{N} \hat{z}_{nk} \qquad \hat{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \hat{z}_{nk} \boldsymbol{x}_n$$

$$\hat{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} \hat{z}_{nk} (\boldsymbol{x}_n - \hat{\mu}_k)(\boldsymbol{x}_n - \hat{\mu}_k)^{\top}$$

$N_k$ : Effective number of points in cluster k

Does that matter? Should we worry that we aren't solving the actual problem anymore?

Not really; will see the justification soon ☺

But wait! This is not the same as $\sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\Theta)$ which was the original MLE objective for this LVM ☹

# Expectation-Maximization (EM) for GMM

Expectation of CLL

.. which we maximized in ALT-OPT

- EM finds $\Theta_{MLE}$ by maximizing $\mathbb{E}[\log p(\boldsymbol{X}, \boldsymbol{Z}|\Theta)]$ rather than $\log p(\boldsymbol{X}, \hat{\boldsymbol{Z}}|\Theta)$
- Note: Expectation will be w.r.t. the <u>conditional</u> posterior distribution of $\boldsymbol{Z}$, i.e., $p(\boldsymbol{Z}|\boldsymbol{X}, \Theta)$

It is "conditional" posterior because it is also conditioned on $\Theta$, not just data $X$

Why w.r.t. this distribution? Will see justification in a bit

- The EM algorithm for GMM operates as follows
  - Initialize $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$ as $\hat{\Theta}$
  - Repeat until convergence

Needed to get the expected CLL

Requires knowing $\Theta$

    - Compute conditional posterior $p(\boldsymbol{Z}|\boldsymbol{X}, \hat{\Theta})$. Since obs are i.i.d, compute separately for each $n$ (and for $k = 1,2,..K$)

Same as $p(z_{nk} = 1|\boldsymbol{x}_n, \hat{\Theta})$, just a different notation

$$p(\boldsymbol{z}_n = k|\boldsymbol{x}_n, \hat{\Theta}) \propto p(\boldsymbol{z}_n = k|\hat{\Theta})\, p(\boldsymbol{x}_n|\boldsymbol{z}_n = k, \hat{\Theta}) = \hat{\pi}_k \mathcal{N}(x_n|\hat{\mu}_k, \hat{\Sigma}_k)$$

  - Update $\Theta$ by maximizing the expected complete data log-likelihood

Solution has a similar form as ALT-OPT (or gen. class.), except we now have the expectation of $z_{nk}$ being used

$$\hat{\Theta} = \text{argmax}_\Theta \mathbb{E}_{p(\boldsymbol{Z}|\boldsymbol{X}, \hat{\Theta})}[\log p(\boldsymbol{X}, \boldsymbol{Z}|\Theta)] = \sum_{n=1}^N \mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n, \hat{\Theta})}[\log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\Theta)]$$

$$= \text{argmax}_\Theta\, \mathbb{E}\left[\sum_{n=1}^N \sum_{k=1}^K z_{nk}[\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)]\right]$$

$$= \text{argmax}_\Theta \sum_{n=1}^N \sum_{k=1}^K \mathbb{E}[z_{nk}][\log \pi_k + \log \mathcal{N}(\boldsymbol{x}_n|\mu_k, \Sigma_k)]$$

$$\hat{\pi}_k = \frac{1}{N}\sum_{n=1}^N \mathbb{E}[z_{nk}] \qquad \hat{\mu}_k = \frac{1}{N_k}\sum_{n=1}^N \mathbb{E}[z_{nk}]\boldsymbol{x}_n$$

$N_k$ : Effective number of points in cluster k

$$\hat{\Sigma}_k = \frac{1}{N_k}\sum_{n=1}^N \mathbb{E}[z_{nk}](\boldsymbol{x}_n - \hat{\mu}_k)(\boldsymbol{x}_n - \hat{\mu}_k)^\top$$

# EM for GMM (Contd)



Reason: $\sum_{k=1}^{K} \gamma_{nk} = 1$

Need to normalize: $\mathbb{E}[z_{nk}] = \frac{\hat{\pi}_k \mathcal{N}(x_n|\hat{\mu}_k, \hat{\Sigma}_k)}{\sum_{\ell=1}^{K} \hat{\pi}_\ell \mathcal{N}(x_n|\hat{\mu}_\ell, \hat{\Sigma}_\ell)}$

- The EM algo for GMM required $\mathbb{E}[z_{nk}]$. Note $z_{nk} \in \{0,1\}$

$$\mathbb{E}[z_{nk}] = \gamma_{nk} = 0 \times p(z_{nk} = 0|x_n, \hat{\Theta}) + 1 \times p(z_{nk} = 1|x_n, \hat{\Theta}) = p(z_{nk} = 1|x_n, \hat{\Theta}) \propto \hat{\pi}_k \mathcal{N}(x_n|\hat{\mu}_k, \hat{\Sigma}_k)$$

## EM for Gaussian Mixture Model

1. Initialize $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ as $\Theta^{(0)}$, set $t = 1$
2. E step: compute the expectation of each $\boldsymbol{z}_n$ (we need it in M step)

Accounts for fraction of points in each cluster

Accounts for cluster shapes (since each cluster is a Gaussian

Soft K-means, which are more of a heuristic to get soft-clustering, also gave us probabilities but didn't account for cluster shapes or fraction of points in each cluster

$$\mathbb{E}[z_{nk}^{(t)}] = \gamma_{nk}^{(t)} = \frac{\pi_k^{(t-1)} \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k^{(t-1)}, \boldsymbol{\Sigma}_k^{(t-1)})}{\sum_{\ell=1}^{K} \pi_\ell^{(t-1)} \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_\ell^{(t-1)}, \boldsymbol{\Sigma}_\ell^{(t-1)})} \quad \forall n, k$$

3. Given "responsibilities" $\gamma_{nk} = \mathbb{E}[z_{nk}]$, and $N_k = \sum_{n=1}^{N} \gamma_{nk}$, re-estimate $\Theta$ via MLE

Effective number of points in the $k^{th}$ cluster

M-step:

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

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

$$\pi_k^{(t)} = \frac{N_k}{N}$$

4. Set $t = t + 1$ and go to step 2 if not yet converged