Machine Learning for Data Science (CS4786)
Lecture 14

Latent Variables, EM Algorithm

Course Webpage :
http://www.cs.cornell.edu/Courses/cs4786/2016sp/

Examples

- Gaussian Mixture Model
  - Each θ consists of mixture distribution $\pi = (\pi_1, \ldots, \pi_K)$, means $\mu_1, \ldots, \mu_K \in \mathbb{R}^d$ and covariance matrices $\Sigma_1, \ldots, \Sigma_K$
  - At time t we generate a new tree as follows:

$$c_t \sim \pi, \qquad x_t \sim N(\mu_{c_t}, \Sigma_{c_t})$$

each theta consists
μ1-μk: tree location of center
π = (0.2, 0.1, 0.7)μ
u -> u1 u2 u3
covariance matrices ∑1, ∑2, ∑3

Probabilistic Models

More generally:

- Θ consists of set of possible parameters
- We have a distribution P_θ over the data induced by each $\theta \in \Theta$
- Data is generated by one of the $\theta \in \Theta$
- Learning: Estimate value or distribution for $\theta^* \in \Theta$ given data

Maximum Likelihood Principal

Pick $\theta \in \Theta$ that maximizes probability of observation

$$\theta_{MLE} = \text{argmax}_{\theta \in \Theta} \underbrace{\log P_\theta(x_1, \ldots, x_n)}_{\text{Likelihood}}$$

Maximum A Posteriori

Pick $\theta \in \Theta$ that is most likely given data

$$
\begin{aligned}
\theta_{MAP} &= \operatorname{argmax}_{\theta \in \Theta} \log P(\theta | x_1, \ldots, x_n) \\
&= \operatorname{argmax}_{\theta \in \Theta} \log \left(\underbrace{P(x_1, \ldots, x_n | \theta)}_{\text{likelihood}} \underbrace{P(\theta)}_{\text{prior}} \right) \\
&= \operatorname{argmax}_{\theta \in \Theta} \underset{\text{log likelihood}}{\log \left(P(x_1, \ldots, x_n | \theta) \right)} + \underset{\text{log prior}}{\log \left(P(\theta) \right)}
\end{aligned}
$$

Example: Gaussian Mixture Model

MLE: $\theta = (\mu_1, \ldots, \mu_K), \pi, \Sigma$

$$P_\theta(x_1, \ldots, x_n) = \prod_{t=1}^{n} \left(\sum_{i=1}^{K} \pi_i \frac{1}{\sqrt{(2 * 3.1415)^2 |\Sigma_i|}} \exp\left(-(x_t - \mu_i)^\top \Sigma_i (x_t - \mu_i) \right) \right)$$

Find θ that maximizes $\log P_\theta(x_1, \ldots, x_n)$

MLE for GMM

Let us consider the one dimensional case,

$$\log P_\theta(x_{1,\ldots,n}) = \sum_{t=1}^{n} \log\left(\sum_{i=1}^{K} \pi_i \frac{1}{\sqrt{2 * 3.1415\sigma_i^2}} \exp\left(-(x_t - \mu_i)^2/\sigma_i^2\right)\right)$$

Now consider the partial derivative w.r.t. μ_1, we have:

$$\frac{\partial \log P_\theta(x_{1,\ldots,n})}{\partial \mu_1} = \sum_{t=1}^{n} \frac{\frac{\pi_1}{\sigma_1} \exp\left(-\frac{(x_t-\mu_1)^2}{\sigma_1^2}\right)}{\sum_{i=1}^{K} \frac{\pi_i}{\sigma_i} \exp\left(-\frac{(x_t-\mu_i)^2}{\sigma_i^2}\right)}$$

Even given all other parameters, optimizing w.r.t. just μ_1 is hard!

MLE for GMM

Say by some magic you knew cluster assignments, then

$$\log P_\theta((x_t, c_t)_{1,\dots,n}) = \sum_{t=1}^{n} \log\left(\frac{\pi_{c_t}}{\sqrt{2 * 3.1415\sigma_{c_t}^2}} \exp\left(-\frac{(x_t - \mu_{c_t})^2}{2\sigma_{c_t}^2}\right)\right)$$

$$= \sum_{t=1}^{n}\left(\log(\pi_{c_t}) - \log(2 * 3.1415 * \sigma_{c_t}^2) - \frac{(x_t - \mu_{c_t})^2}{2\sigma_{c_t}^2}\right)$$

Now consider the partial derivative w.r.t. μ_i, we have:

$$\frac{\partial \log P_\theta((x_t, c_t)_{1,\dots,n})}{\partial \mu_i} = -\frac{\partial}{\partial \mu_i} \sum_{t=1}^{n}\left(\frac{1}{2\sigma_{c_t}^2}(x_t - \mu_{c_t})^2\right)$$

$$= -\frac{1}{2\sigma_i^2}\frac{\partial}{\partial \mu_i}\sum_{t:c_t=i}(x_t - \mu_i)^2$$

$$= \frac{1}{\sigma_i^2}\sum_{t:c_t=i}(x_t - \mu_i)$$

Latent Variables

- We only observe $x_1, \ldots, x_n$, cluster assignments $c_1, \ldots, c_n$ are not observed
- Finding $\theta \in \Theta$ (even for 1-d GMM) that directly maximizes Likelihood or A Posteriori given $x_1, \ldots, x_n$ is hard!
- Given latent variables $c_1, \ldots, c_n$, the problem of maximizing likelihood (or a posteriori) became easy

Can we use latent variables to device an algorithm?

Expectation Maximization Algorithm

- For demonstration we shall consider the problem of finding MLE (MAP version is very similar)

Expectation Maximization Algorithm

- For demonstration we shall consider the problem of finding MLE (MAP version is very similar)
- Initialize $\theta^{(0)}$ arbitrarily, repeat unit convergence:

(E step) For every t, define distribution Q_t over the latent variable c_t as:

$$Q_t^{(i)}(c_t) = P(c_t|x_t, \theta^{(i-1)})$$

(M step)

$$\theta^{(i)} = \operatorname{argmax}_{\theta \in \Theta} \sum_{t=1}^{n} \sum_{c_t} Q_t^{(i)}(c_t) \log P(x_t, c_t|\theta)$$

Example: EM for GMM

- E step: For every $k \in [K]$,

$$Q_t^{(i)}(c_t = k) = P\left(c_t = k | x_t, \theta^{(i-1)}\right) = P\left(x_t | c_t = k, \theta^{(i-1)}\right) \times P\left(c_t = k | \theta^{(i-1)}\right)$$
$$\propto \underbrace{\phi\left(x_t; \mu_k^{(i-1)}, \Sigma_k^{(i-1)}\right)}_{\text{gaussian p.d.f.}} \times \pi_k^{(i-1)}$$

Example: EM for GMM

- E step: For every $k \in [K]$,

$$Q_t^{(i)}(c_t = k) = P\left(c_t = k | x_t, \theta^{(i-1)}\right) = P\left(x_t | c_t = k, \theta^{(i-1)}\right) \times P\left(c_t = k | \theta^{(i-1)}\right)$$
$$\propto \underbrace{\phi\left(x_t; \mu_k^{(i-1)}, \Sigma_k^{(i-1)}\right)}_{\text{gaussian p.d.f.}} \times \pi_k^{(i-1)}$$

- M step: Given $Q_1, \ldots, Q_n$, we need to find

$$\begin{aligned}
\theta^{(i)} &= \underset{\theta \in \Theta}{\operatorname{argmax}} \sum_{t=1}^{n} \sum_{k=1}^{K} Q_t^{(i)}(k) \log P(x_t, c_t = k | \theta) \\
&= \underset{\theta}{\operatorname{argmax}} \sum_{t=1}^{n} \sum_{k=1}^{K} Q_t^{(i)}(k) \left(\log P(x_t | c_t = k, \theta) + \log P(c_t = k | \theta)\right) \\
&= \underset{\pi, \mu_{1,\ldots,K}, \Sigma_{1,\ldots,K}}{\operatorname{argmax}} \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q_t^{(i)}(k) \left(\log \phi\left(x_t; \mu_k, \Sigma_k\right) + \log \pi_k\right)
\end{aligned}$$

Example: EM for GMM

For every $k \in [K]$, the maximization step yields,

$$\mu_k^{(i)} = \frac{\sum_{t=1}^{n} Q_t^{(i)}(k) x_t}{\sum_{t=1}^{n} Q_t(k)}, \quad \Sigma_k^{(i)} = \frac{\sum_{t=1}^{n} Q_t^{(i)}(k) \left(x_t - \mu_k^{(i)}\right)\left(x_t - \mu_k^{(i)}\right)^\top}{\sum_{t=1}^{n} Q_t(k)}$$

$$\pi_k^{(i)} = \frac{\sum_{t=1}^{n} Q_t^{(i)}(k)}{n}$$

Why should EM work?

A very high level view:

- Performing E-step will never decrease log-likelihood (or log a posteriori)

Why should EM work?

A very high level view:

- Performing E-step will never decrease log-likelihood (or log a posteriori)
- Performing M-step will never decrease log-likelihood (or log a posteriori)

Why should EM work?

Steps to show that $\log \mathrm{Lik}(\theta^{(i+1)}) \geq \log \mathrm{Lik}(\theta^{(i)})$:

$$\begin{aligned}
\log P_{\theta^{(i+1)}}(x_1, \ldots, x_n) &= \sum_{t=1}^{n} \log P_{\theta^{(i+1)}}(x_t) \\
&\geq \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q^{(i)}(c_t) \log\left(\frac{P_{\theta^{(i+1)}}(x_t, c_t)}{Q^{(i)}(c_t)}\right) \\
&\geq \sum_{t=1}^{n} \sum_{c_t=1}^{K} Q^{(i)}(c_t) \log\left(\frac{P_{\theta^{(i)}}(x_t, c_t)}{Q^{(i)}(c_t)}\right) \\
&= \sum_{t=1}^{n} \log P_{\theta^{(i)}}(x_t)
\end{aligned}$$

Why should EM work?

- Likelihood never decreases
- So whenever we converge we converge to a local optima
- However problem is non-convex and can have many local optimal
- In general no guarantee on rate of convergence
- In practice, do multiple random initializations and pick the best one!

EM IN GENERAL

- There was nothing special about GMM or clustering problems
- EM can be used as a general strategy for any problem with latent/missing/unobserved variables
- The MAP version only involves an extra prior term over θ multiplied to the likelihood
- In general probabilistic models with observed and latent variables can be represented succinctly as graphical models.
  Next time ...