Expectation maximization

Subhransu Maji

CMPSCI 689: Machine Learning

14 April 2015

Motivation

- Suppose you are building a naive Bayes spam classifier. After your are done your boss tells you that there is no money to label the data.
  - You have a probabilistic model that assumes labelled data, but you don't have any labels. Can you still do something?

- Amazingly you can!
  - Treat the labels as hidden variables and try to learn them simultaneously along with the parameters of the model

- Expectation Maximization (EM)
  - A broad family of algorithms for solving hidden variable problems
  - In today's lecture we will derive EM algorithms for clustering and naive Bayes classification and learn why EM works

Gaussian mixture model for clustering

- Suppose data comes from a Gaussian Mixture Model (GMM) — you have K clusters and the data from the cluster k is drawn from a Gaussian with mean μ_k and variance σ_k^2
- We will assume that the data comes with labels (we will soon remove this assumption)
- Generative story of the data:
  - For each example $n = 1, 2, .., N$
    - Choose a label $y_n \sim \text{Mult}(\theta_1, \theta_2, \ldots, \theta_K)$
    - Choose example $\mathbf{x}_n \sim N(\mu_k, \sigma_k^2)$
- Likelihood of the data:

$$p(\mathcal{D}) = \prod_{n=1}^{N} p(y_n)p(\mathbf{x}_n|y_n) = \prod_{n=1}^{N} \theta_{y_n} N(\mathbf{x}_n; \mu_{y_n}, \sigma_{y_n}^2)$$

$$p(\mathcal{D}) = \prod_{n=1}^{N} \theta_{y_n} \left(2\pi\sigma_{y_n}^2\right)^{-\frac{D}{2}} \exp\left(-\frac{||\mathbf{x}_n - \mu_{y_n}||^2}{2\sigma_{y_n}^2}\right)$$

GMM: known labels

- Likelihood of the data:

$$p(\mathcal{D}) = \prod_{n=1}^{N} \theta_{y_n} \left(2\pi\sigma_{y_n}^2\right)^{-\frac{D}{2}} \exp\left(-\frac{||\mathbf{x}_n - \mu_{y_n}||^2}{2\sigma_{y_n}^2}\right)$$

- If you knew the labels y_n then the maximum-likelihood estimates of the parameters is easy:

$$\theta_k = \frac{1}{N}\sum_n [y_n = k]$$

fraction of examples with label k

$$\mu_k = \frac{\sum_n [y_n = k]\mathbf{x}_n}{\sum_n [y_n = k]}$$

mean of all the examples with label k

$$\sigma_k^2 = \frac{\sum_n [y_n = k]||\mathbf{x}_n - \mu_k||^2}{\sum_n [y_n = k]}$$

variance of all the examples with label k

GMM: unknown labels

- Now suppose you didn't have labels y_n. Analogous to k-means, one solution is to iterate. Start by guessing the parameters and then repeat the two steps:
  - Estimate labels given the parameters
  - Estimate parameters given the labels

- In k-means we assigned each point to a single cluster, also called as hard assignment (point 10 goes to cluster 2)
- In expectation maximization (EM) we will will use soft assignment (point 10 goes half to cluster 2 and half to cluster 5)

- Lets define a random variable $\mathbf{z}_n = [z_1, z_2, \ldots, z_K]$ to denote the assignment vector for the n^{th} point
  - Hard assignment: only one of z_k is 1, the rest are 0
  - Soft assignment: z_k is positive and sum to 1

GMM: parameter estimation

- Formally $z_{n,k}$ is the probability that the n^{th} point goes to cluster k

$$z_{n,k} = p(y_n = k|\mathbf{x}_n)$$
$$= \frac{P(y_n = k, \mathbf{x}_n)}{P(\mathbf{x_n})}$$
$$\propto P(y_n = k)P(\mathbf{x}_n|y_n) = \theta_k N(\mathbf{x}_n; \mu_k, \sigma_k^2)$$

- Given a set of parameters $(\theta_k, \mu_k, \sigma_k^2)$, $z_{n,k}$ is easy to compute
- Given $z_{n,k}$, we can update the parameters $(\theta_k, \mu_k, \sigma_k^2)$ as:

$$\theta_k = \frac{1}{N}\sum_n z_{n,k}$$

fraction of examples with label k

$$\mu_k = \frac{\sum_n z_{n,k}\mathbf{x}_n}{\sum_n z_{n,k}}$$

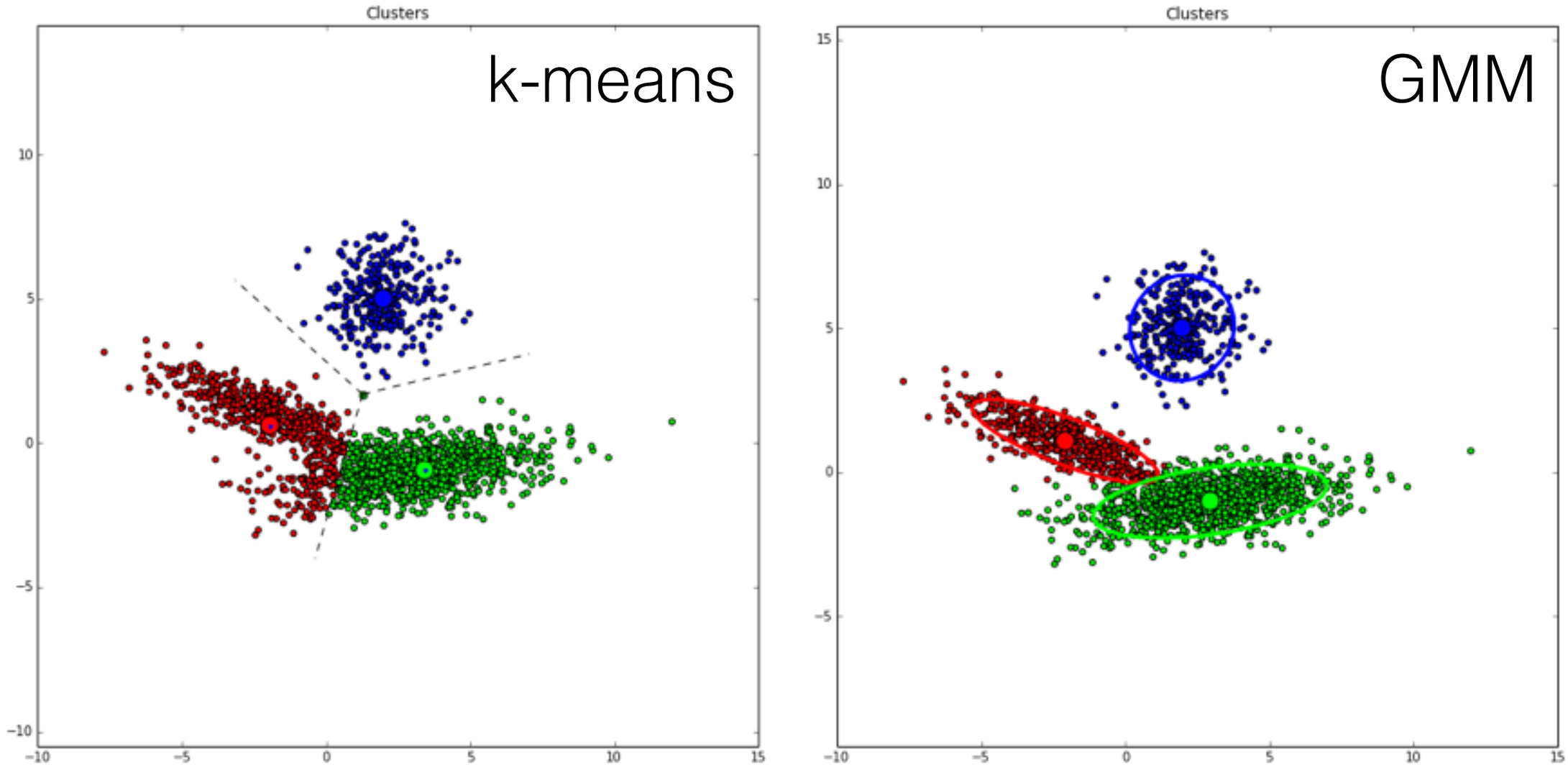
mean of all the fractional examples with label k

$$\sigma_k^2 = \frac{\sum_n z_{n,k}||\mathbf{x}_n - \mu_k||^2}{\sum_n z_{n,k}}$$

variance of all the fractional examples with label k

GMM: example

- We have replaced the indicator variable $[y_n = k]$ with $p(y_n=k)$ which is the expectation of $[y_n=k]$. This is our guess of the labels.
- Just like k-means the EM is susceptible to local minima.
- Clustering example:



http://nbviewer.ipython.org/github/NICTA/MLSS/tree/master/clustering/

The EM framework

- We have data with observations x_n and hidden variables y_n, and would like to estimate parameters θ
- The likelihood of the data and hidden variables:

$$p(\mathcal{D}) = \prod_n p(\mathbf{x}_n, y_n|\theta)$$

- Only x_n are known so we can compute the data likelihood by marginalizing out the y_n:

$$p(\mathbf{X}|\theta) = \prod_n \sum_{y_n} p(\mathbf{x}_n, y_n|\theta)$$

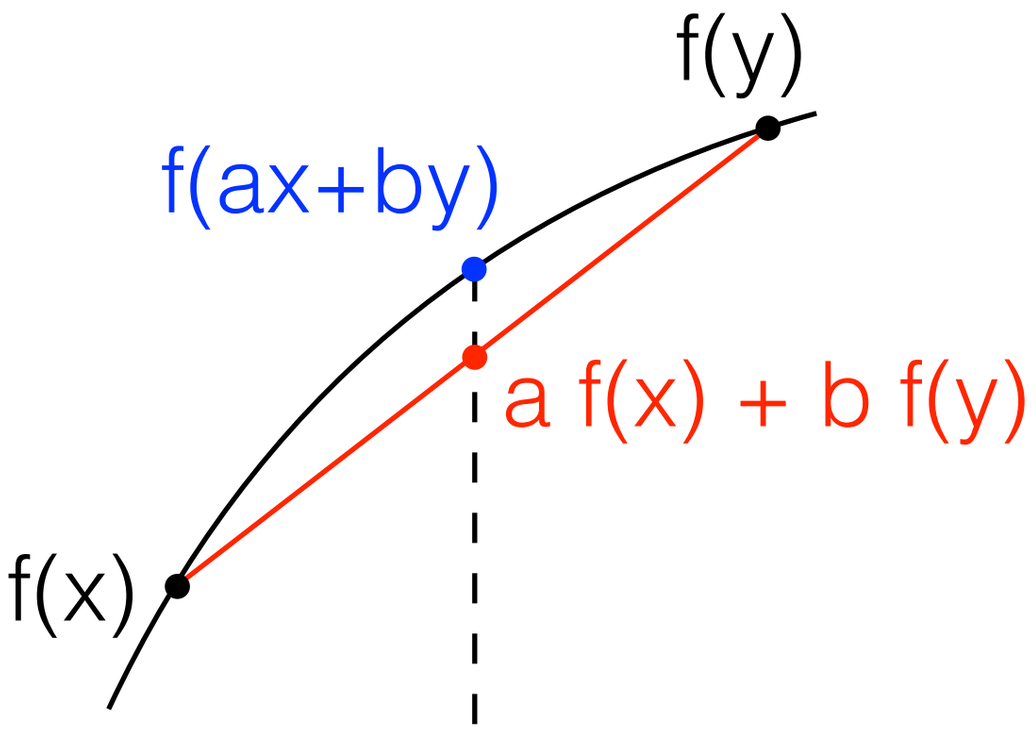
- Parameter estimation by maximizing log-likelihood:

$$\theta_{\mathrm{ML}} \leftarrow \arg\max_\theta \sum_n \log \left(\sum_{y_n} p(\mathbf{x}_n, y_n|\theta) \right)$$

hard to maximize since the sum is inside the log

Jensen's inequality

- Given a concave function f and a set of weights $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$
- Jensen's inequality states that $f(\sum_i \lambda_i x_i) \geq \sum_i \lambda_i f(x_i)$
- This is a direct consequence of concavity
  - $f(ax + by) \geq a f(x) + b f(y)$ when $a \geq 0$, $b \geq 0$, $a + b = 1$



The EM framework

- Construct a lower bound the log-likelihood using Jensen's inequality

$$\mathcal{L}(\mathbf{X}|\theta) = \sum_n \log \left(\sum_{y_n} p(\mathbf{x}_n, y_n|\theta) \right)$$

$$= \sum_n \overset{\text{f}}{\uparrow} \log \left(\sum_{y_n} q(y_n) \frac{p(\mathbf{x}_n, y_n|\theta)}{q(y_n)} \right)$$

X

λ

Jensen's inequality

$$\geq \sum_n \sum_{y_n} q(y_n) \log \left(\frac{p(\mathbf{x}_n, y_n|\theta)}{q(y_n)} \right)$$

$$= \sum_n \sum_{y_n} [q(y_n) \log p(\mathbf{x}_n, y_n|\theta) - q(y_n) \log q(y_n)]$$

$$\triangleq \hat{\mathcal{L}}(\mathbf{X}|\theta)$$

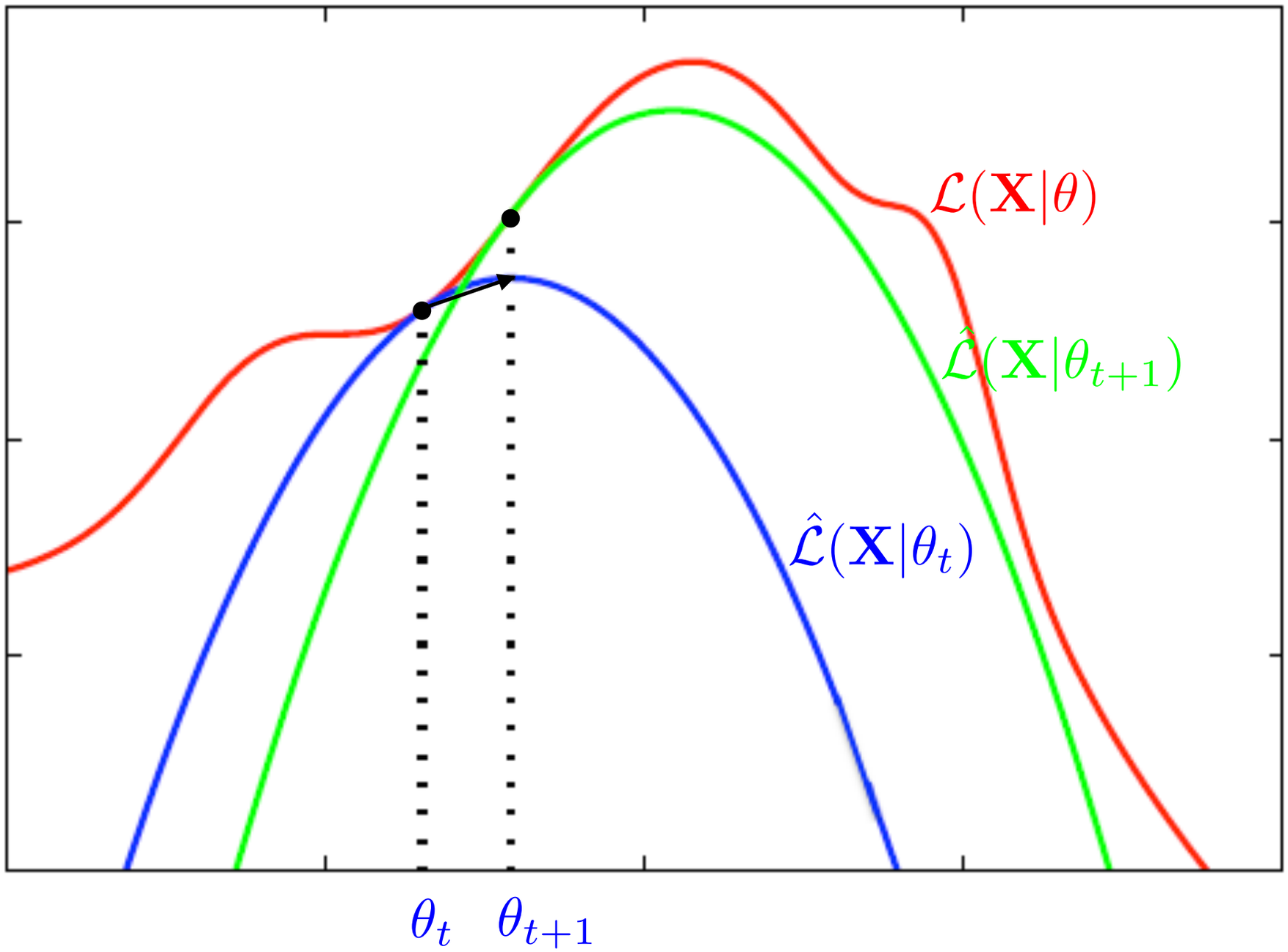
independent of θ

- Maximize the lower bound:

$$\theta \leftarrow \arg\max_\theta \sum_n \sum_{y_n} q(y_n) \log p(\mathbf{x}_n, y_n|\theta)$$

Lower bound illustrated

- Maximizing the lower bound increases the value of the original function if the lower bound touches the function at the current value



An optimal lower bound

- Any choice of the probability distribution $q(y_n)$ is valid as long as the lower bound touches the function at the current estimate of θ

$$\mathcal{L}(\mathbf{X}|\theta_t) = \hat{\mathcal{L}}(\mathbf{X}|\theta_t)$$

- We can the pick the optimal $q(y_n)$ by maximizing the lower bound

$$\arg\max_q \sum_{y_n} [q(y_n) \log p(\mathbf{x}_n, y_n|\theta) - q(y_n) \log q(y_n)]$$

- This gives us $q(y_n) \leftarrow p(y_n|\mathbf{x}_n, \theta_t)$
  - Proof: use Lagrangian multipliers with "sum to one" constraint

- This is the distributions of the hidden variables conditioned on the data and the current estimate of the parameters
  - This is exactly what we computed in the GMM example

The EM algorithm

- We have data with observations x_n and hidden variables y_n, and would like to estimate parameters θ of the distribution $p(x \mid \theta)$
- EM algorithm
  - Initialize the parameters θ randomly
  - Iterate between the following two steps:
    - ➡ E step: Compute probability distribution over the hidden variables

$$q(y_n) \leftarrow p(y_n|\mathbf{x}_n, \theta)$$

    - ➡ M step: Maximize the lower bound

$$\theta \leftarrow \arg\max_{\theta} \sum_{n} \sum_{y_n} q(y_n) \log p(\mathbf{x}_n, y_n|\theta)$$

- EM algorithm is a great candidate when M-step can done easily but $p(x \mid \theta)$ cannot be easily optimized over θ
  - For e.g. for GMMs it was easy to compute means and variances given the memberships

Naive Bayes: revisited

- Consider the binary prediction problem
- Let the data be distributed according to a probability distribution:

$$p_\theta(y, \mathbf{x}) = p_\theta(y, x_1, x_2, \ldots, x_D)$$

- We can simplify this using the chain rule of probability:

$$p_\theta(y, \mathbf{x}) = p_\theta(y) p_\theta(x_1|y) p_\theta(x_2|x_1, y) \ldots p_\theta(x_D|x_1, x_2, \ldots, x_{D-1}, y)$$

$$= p_\theta(y) \prod_{d=1}^{D} p_\theta(x_d|x_1, x_2, \ldots, x_{d-1}, y)$$

- **Naive Bayes** assumption:

$$p_\theta(x_d|x_{d'}, y) = p_\theta(x_d|y), \forall d' \neq d$$

- E.g., The words "free" and "money" are independent given spam

Naive Bayes: a simple case

- **Case:** binary labels and binary features

$$\left.\begin{aligned} p_\theta(y) &= \texttt{Bernoulli}(\theta_0) \\ p_\theta(x_d|y=1) &= \texttt{Bernoulli}(\theta_d^+) \\ p_\theta(x_d|y=-1) &= \texttt{Bernoulli}(\theta_d^-) \end{aligned}\right\} \text{1+2D parameters}$$

- Probability of the data:

$$\begin{aligned} p_\theta(y, \mathbf{x}) &= p_\theta(y) \prod_{d=1}^{D} p_\theta(x_d|y) \\ &= \theta_0^{[y=+1]}(1-\theta_0)^{[y=-1]} \\ &\ldots \times \prod_{d=1}^{D} \theta_d^{+[x_d=1,y=+1]}(1-\theta_d^+)^{[x_d=0,y=+1]} \quad \text{// label +1} \\ &\ldots \times \prod_{d=1}^{D} \theta_d^{-[x_d=1,y=-1]}(1-\theta_d^-)^{[x_d=0,y=-1]} \quad \text{// label -1} \end{aligned}$$

Naive Bayes: parameter estimation

- Given data we can estimate the parameters by maximizing data likelihood
- The maximum likelihood estimates are:

$$\hat{\theta}_0 = \frac{\sum_n [y_n = +1]}{N}$$

// fraction of the data with label as +1

$$\hat{\theta}_d^+ = \frac{\sum_n [x_{d,n} = 1, y_n = +1]}{\sum_n [y_n = +1]}$$

// fraction of the instances with 1 among +1

$$\hat{\theta}_d^- = \frac{\sum_n [x_{d,n} = 1, y_n = -1]}{\sum_n [y_n = -1]}$$

// fraction of the instances with 1 among -1

Naive Bayes: EM

- Now suppose you don't have labels y_n
- Initialize the parameters θ randomly
- E step: compute the distribution over the hidden variables $q(y_n)$

$$q(y_n = 1) = p(y_n = +1 | \mathbf{x}_n, \theta) \propto \theta_0^+ \prod_{d=1}^{D} \theta_d^{+[x_{d,n}=1]} (1 - \theta_d^+)^{[x_{d,n}=0]}$$

- M step: estimate θ given the guesses

$$\theta_0 = \frac{\sum_n q(y_n = 1)}{N}$$ // fraction of the data with label as +1

$$\theta_d^+ = \frac{\sum_n [x_{d,n} = 1] q(y_n = 1)}{\sum_n q(y_n = 1)}$$ // fraction of the instances with 1 among +1

$$\theta_d^- = \frac{\sum_n [x_{d,n} = 1] q(y_n = -1)}{\sum_n q(y_n = -1)}$$ // fraction of the instances with 1 among -1

Summary

- **Expectation maximization**
  - A general technique to estimate parameters of probabilistic models when some observations are hidden
  - EM iterates between estimating the hidden variables and optimizing parameters given the hidden variables
  - EM can be seen as a maximization of the lower bound of the data log-likelihood — we used Jensen's inequality to switch the log-sum to sum-log
- **EM can be used for learning:**
  - mixtures of distributions for clustering, e.g. GMM
  - parameters for hidden Markov models (next lecture)
  - topic models in NLP
  - probabilistic PCA
  - ....

Slides credit

- Some of the slides are based on CIML book by Hal Daume III
- The figure for the EM lower bound is based on https://cxwangyi.wordpress.com/2008/11/
- Clustering k-means vs GMM is from http://nbviewer.ipython.org/github/NICTA/MLSS/tree/master/clustering/