EM
CS B553
Spring 2013

Announcements

• A4 released
  – Due Sunday April 28
  – Markov chains plus a little learning
• Final projects
  – Presentations next week
  – Report due Wednesday May 1
• Course evaluations
  – Online – please complete these! (Important for deciding whether to continue offering the course.)

Final project presentations

• Should summarize the “story” of your project
  – Introduction, background, motivation
  – Your approach to the problem
  – (Preliminary) experimental results
• Schedule of presentations posted on wiki
  – Please check to make sure you’re on the schedule...
  – 2-person groups next Tuesday, 8 minutes each
  – 1-person groups next Thursday, 5 minutes each

Learning with unobserved variables

• So far, we’ve assumed that our exemplars for learning include values for all variables
• What if we can’t observe all variables?
  – Given a network G and form for the CPDs, estimate \( \hat{\theta} \), i.e. find,
    \[
    \theta^* = \arg \max_{\theta} P(\theta|D)
    \]
• Schedule of presentations posted on wiki
  – Please check to make sure you’re on the schedule...
  – 2-person groups next Tuesday, 8 minutes each
  – 1-person groups next Thursday, 5 minutes each

A chicken and egg problem!

• If we knew the model \( \theta \), then we could guess the unknown values in the exemplars
  – Or marginalize them out
• If we knew the unknown values, then we could estimate the model \( \theta \)

One approach

• Estimate an initial model, and then compute distributions over unknown variable values
  – E.g. Initial model:
    \[
    \begin{align*}
    \theta_{a_1} &= 0.3 & \theta_{a_2} &= 0.9 \\
    \theta_{b_1|a_1} &= 0.1 & \theta_{b_1|a_2} &= 0.8 \\
    \theta_{c_1|a_1,b_1} &= 0.83 & \theta_{c_1|a_2,b_1} &= 0.6 \\
    \theta_{c_1|a_1,b_2} &= 0.09 & \theta_{c_1|a_2,b_2} &= 0.2.
    \end{align*}
    \]
  – For exemplar \( \alpha=(a_1, ?, ?, d_0) \), compute \( P(B,C|\alpha, \theta) \):
    \[
    \begin{align*}
    Q((b^1, c^1)) &= 0.3 \cdot 0.9 \cdot 0.2 \cdot 0.2/0.2196 = 0.0492 \\
    Q((b^2, c^1)) &= 0.3 \cdot 0.9 \cdot 0.8 \cdot 0.9/0.2196 = 0.8852 \\
    Q((b^1, c^2)) &= 0.3 \cdot 0.1 \cdot 0.6 \cdot 0.2/0.2196 = 0.0164 \\
    Q((b^2, c^2)) &= 0.3 \cdot 0.1 \cdot 0.4 \cdot 0.0/0.2196 = 0.0002.
    \end{align*}
    \]
One approach

• Given this distribution over unknown variables:
  \[ Q(\theta, \phi) = 0.3 \cdot 0.9 \cdot 0.2 \cdot 0.2 / 0.2196 = 0.0492 \]
  \[ Q(\theta, \phi) = 0.3 \cdot 0.9 \cdot 0.8 \cdot 0.9 / 0.2196 = 0.8852 \]
  \[ Q(\theta, \phi) = 0.3 \cdot 0.1 \cdot 0.6 \cdot 0.2 / 0.2196 = 0.0164 \]
  \[ Q(\theta, \phi) = 0.3 \cdot 0.1 \cdot 0.4 \cdot 0.9 / 0.2196 = 0.0492 \]
  • Pretend as though we had four exemplars for the single exemplar \((a1, ?, ?, d0)\)
    - Each one weighted by the corresponding probability above
    - Then re-estimate \(\theta\) using these new exemplars

Expectation-Maximization

• EM is a general technique for learning when some variables are not observable
  • Consists of two steps, run iteratively:
    - \(E\) Step: Use current parameter estimates \(\theta\) to compute expected ("virtual") counts for the number of samples having some property
      (i.e., compute marginals over unknown variables given known variables and current model estimate)
    - \(M\) Step: Use Maximum Likelihood to estimate new parameters, \(\theta^{t+1}\), given these virtual counts

Example: Gaussian Mixture Model

• Useful for data that can be modeled by sum of a small number of Gaussians,
  \[ p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \sigma_k) \]

Example: Gaussian Mixture Model

• Useful for data that can be modeled by sum of a small number of Gaussians,
  \[ p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \sigma_k) \]
  — How to learn parameters from training data with MLE?

Cost function for fitting a GMM

For a point \(x_i\)
  \[ p(x_i) = \sum_{k=1}^{K} \pi_k N(x_i|\mu_k, \sigma_k) \]

The likelihood of the GMM for \(N\) points (assuming independence) is
  \[ \prod_{i=1}^{N} p(x_i) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k N(x_i|\mu_k, \sigma_k) \]

and the (negative) log-likelihood is
  \[ L(\theta) = -\sum_{i=1}^{N} \ln \sum_{k=1}^{K} \pi_k N(x_i|\mu_k, \sigma_k) \]

where \(\theta\) are the parameters we wish to estimate (i.e. \(\mu_k\) and \(\sigma_k\) in this case).

C. Bishop
Expectation Maximization (EM) Algorithm

Step 1 Expectation: Compute responsibilities using current parameters $\mu_k, \tau_k$ (assignment)
$$
\gamma_{ik} = \frac{\pi_k \mathcal{N}(x_i | \mu_k, \tau_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_i | \mu_j, \tau_j)}
$$

Step 2 Maximization: Re-estimate parameters using computed responsibilities
$$
\mu_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} x_i
$$
$$
\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{N} \gamma_{ik} (x_i - \mu_k)(x_i - \mu_k)^T
$$
$$
\pi_k = \frac{N_k}{N}
$$
where
$$
N_k = \sum_{i=1}^{N} \gamma_{ik}
$$
Repeat until convergence.

Practical issues

- How to initialize models?
- How to choose $K$?
- Will it converge?

Hard EM

- Hard EM (or MAP-EM or Max-EM) computes a MAP estimate of missing latent variables
  - Instead of computing a marginal distribution

- Example: k-means clustering
  - E-step: Assign each cluster to the nearest centroid (aka highest probability one for a Gaussian of unit variance)
  - M-step: Estimate new centroid for set of points in each assigned cluster