Today’s Reading:
- HMS, 9.6, 6.4, 7.3.2, 7.4, 8.4

Today’s Lecture:
- Probabilistic Model-based Clustering
- Descriptive Modeling

Upcoming Due Dates:
- H2 due Thu 3/7
Families of Clustering Algorithms

• Partition-based methods
  – e.g., K-means

• Hierarchical clustering
  – e.g., hierarchical agglomerative clustering

• Probabilistic model-based clustering
  – e.g., mixture models
Probabilistic Model-based Clustering

• Assume a probability model for each component cluster
• Mixture Model:

\[ f(x) = \sum_{k=1}^{K} w_k f_k(x; \theta_k) \]

• where \( f_k \) are component distributions
• components: gaussian, poisson, exponential
Gaussian Mixture Models (GMM)

- K components
- model for each component cluster $N(\mu_k, \sigma_k)$

$$p(x) = \sum_{k=1}^{K} w_k f(x; \mu_k, \sigma_k)$$

much borrowed from Andrew Moore’s Clustering Gaussian Mixtures Lecture notes
GMM cont.

- **Generative Model**
  - choose component with probability $w_k$
  - generate $X \sim N(\mu_k, \sigma_k)$
GMM cont.

• But, we need to learn a model from the data...
• D = \{x(1), \ldots, x(n)\}

First, suppose we know \(w_1, \ldots, w_k\)
\[
P(x \mid w_j, \mu_1, \ldots, \mu_k, \sigma_1, \ldots, \sigma_k) = ?
\]

Next,
\[
P(x \mid \mu_1, \ldots, \mu_k, \sigma_1, \ldots, \sigma_k) = ?
\]

And,
\[
P(D \mid \mu_1, \ldots, \mu_k, \sigma_1, \ldots, \sigma_k) = ?
\]
Look Familiar?

- \( P(D \mid \mu_1, \ldots, \mu_k, \sigma_1, \ldots, \sigma_k) \) gives us the probability for a particular choice of \( \mu_1, \ldots, \mu_k, \sigma_1, \ldots, \sigma_k \)
- How do we choose \( \mu_i, \sigma_i \)?
- Find \( \mu_i \) that maximize the likelihood
- Set

\[
\frac{\partial}{\partial \mu_j} \log P(D \mid \mu_1, \ldots, \mu_k, \sigma_1, \ldots, \sigma_k) = 0
\]

and solve for \( \mu_i \)

- Problem: we have a bunch on non-linear non-analytically-solvable equations
- One solution: gradient descent.... slow
- instead....
Expectation Maximization (EM)

- Dempster, Laird, and Rubin, 1977
- extremely popular recently
- applicable in a wide range of problems
- many uses: hidden markov models,
- basic idea is quite simple...
EM setup

- Let $D = \{x(1), \ldots, x(n)\}$ be $n$ observed data vectors
- Let $H = \{z(1), \ldots, z(n)\}$ be $n$ values of hidden variable $Z$ (these might be the cluster labels)
- Then the log-likelihood of the observed data is
  \[
  l(\theta) = \log p(D \mid \theta) = \log \sum_H p(D, H \mid \theta)
  \]
- *both* $\theta$ and $H$ are unknown
- Let $Q(H)$ be any probability distribution for $H$.
  \[
  l(\theta) = \log \sum_H p(D, H \mid \theta)
  = \log \sum_H Q(H) \frac{p(D, H \mid \theta)}{Q(H)}
  \geq \sum_H Q(H) \log \frac{p(D, H \mid \theta)}{Q(H)}
  = \sum_H Q(H) \log p(D, H \mid \theta) + \sum_H Q(H) \log \frac{1}{Q(H)}
  = F(Q, \theta)
  \]
Jensen’s Inequality

EM Algorithm

- EM algorithm alternates between

  Intuition:
  • In the E-step, we estimate the distribution on the hidden variables, conditioned on a particular setting of the parameter vector $\theta^k$
  • In the M-step, we choose new set of parameters $\theta^{k+1}$ to maximize the expected log-likelihood of observed data

- Maximum for E step:
  $$Q^{k+1} = p(H \mid D, \theta^k)$$

- Maximum for M step:
  $$\theta^{k+1} = \arg \max_{\theta} \sum_H p(H \mid D, \theta^k) \log p(D, H \mid \theta^k)$$
Notes

- Often both the E and M step can be solved in closed form
- Neither the E step nor the M step can decrease the log-likelihood
- Under relatively general conditions the algorithm is guaranteed to converge to a local maximum of log-likelihood
- We must specify a starting point for the algorithm, for example a random choice of $\theta$ or $Q$
- We must specify stopping criteria, or convergence detection
- Computational complexity: number of iterations, time to compute E and M steps
Back to our example...

- We want to fit a normal mixture distribution:

\[
f(x) = \sum_{k=1}^{K} w_k f_k(x; \mu_k, \sigma_k)
\]

where \(w_k\) is prior probability of \(x\) belonging to component \(\theta = (w_1, \ldots, w_k, \mu_1, \ldots, \mu_k, \sigma_1, \ldots, \sigma_k)\) is parameter vector

then probability of data point \(x\) coming from \(k\)th class is:

\[
\hat{P}(k | x) = \frac{w_k f_k(x; \mu_k, \sigma_k)}{f(x)}
\]

E-step:

\[
\hat{\theta}_k = \frac{1}{n} \sum_{i=1}^{n} \hat{P}(k | x(i))\quad \hat{\mu}_k = \frac{1}{n w_k} \sum_{i=1}^{n} \hat{P}(k | x(i))x(i)
\]

\[
\hat{\sigma}_k = \frac{1}{n w_k} \sum_{i=1}^{n} \hat{P}(k | x(i))(x(i) - \mu_k)^2
\]
EM Demo

EM demo

EM Comments

- complexity of EM for multivariate gaussian mixtures with K components: dominated by calculation of K covariance matrices.
  - With p dimensions, $O(Kp^2)$ covariance parameters to be estimated
  - Each requires summing over n data points and cluster weights, leading to $O(Kp^2n)$ per step

- Often times there are large increases in likelihood over first few iteration and then can slowly converge; likelihood as function of iterations not necessarily concave
Likelihood as function of EM Iteration

figure 9.2 in text
Probabilistic Model-based clustering

- model provides full distributional description for each component; we may be able to interpret differences in the distributions
- Given the model, each data point has a K-component vector of probabilities that it belongs to each group (why this is called soft-clustering)
- EM algorithm can be extended for MAP and Bayesian estimation
- Method can be extended to data that are not in p dimensional vector form. mixtures of sequence models, e.g. markov models, others
- key cost: assumption of parametric model
and finally...

how do we choose K?
How to choose K

- Choose K that maximizes likelihood?
- NOT.
- As K is increased, the value of the likelihood at maximum cannot decrease
- Problem of scoring models with different complexities
Model Selection

- First, as we mentioned before, there are two possible goals in constructing a model

- Goal #1: summarization
  - we would like to describe the data as precisely as possible
  - given that our model is comprehensible (subjective) or compact (objective)
  - general approach based on data compression and information-theoretic arguments results in a score function:

\[
\text{score}(\theta, M) = \# \text{ of bits to describe the data given the model} + \text{number of bits to describe the model}
\]

\[
\text{score}(\theta, M) = - \log p(D | \theta, M) - \log p(\theta, M)
\]

- # of bits to transmit the data that is not accounted for by the model
- # of bits to transmit the model
Model Selection, cont.

• Goal #2: generalize from the available data to new data
  – goodness of fit is part of the objective
  – but, since the data is not the entire population, we want to learn a model that will generalize to other new data instances

• In both cases, we want a score function that strikes a compromise between how well the model fits the data and the simplicity of the model, although the theoretical motivations for this are quite different
Bias-Variance

- Model too flexible $\Rightarrow$ overfit the data $\Rightarrow$ high variance
- Model too restrictive $\Rightarrow$ can’t fit the data $\Rightarrow$ high bias
- Bias-variance tradeoff: compromise
Penalize Complexity

- Score(M) = error(M) + penalty-function(M)
- Penalty-function:
  - $M_1, \ldots, M_K$ K models
  - $\theta_k$ best parameters
  - complexity of model $M$ depends on number of parameters $d_k$
- AIC - Akaike information criterion:

  $$S_{\text{AIC}}(M_k) = 2S_L(\hat{\theta}_k; M_k) + 2d_k$$
- BIC – Bayesian information criterion

  $$S_{\text{BIC}}(M_k) = 2S_L(\hat{\theta}_k; M_k) + d_k \log n$$
- Others – Minimum Description Length (MDL), Structural Risk Minimization (SRM)
Bayesian Approach

- compute posterior probability of each model directly and select the one with highest posterior:

\[
p(M_k | D) \propto p(D | M_k)p(M_k) \\
= \int p(D, \theta_k | M_k)p(M_k)d\theta_k \\
= \int p(D |, \theta_k | M_k)p(M_k)d\theta_k
\]

implicitly penalizes complexity, since higher dimensional parameter spaces (more complex models) will mean that the probability mass in \( p(\theta_k | M_k) \) is spread more thinly

often computationally intractable; use MCMC to estimate, use approximation.
External Validation

• General Idea:
  – Split data randomly into two disjoint sets
  – the training set
  – the validation set
  – construct a model based on the training set
  – compute an estimate for the score based on the validation set
  – this estimate for the score will be an unbiased estimate
External Validation, cont.

• popular method: Cross-validation
  – repeatedly split the data into train and validation sets; average results
  – n-fold cross-validation
  – leave one out cross-validation
Summary

• Families of Clustering Algorithms
  – Partition-based methods
  – Hierarchical clustering
  – Probabilistic model-based clustering

• EM Algorithm

• Methods for scoring models with different complexities
Next Time

• Density Estimation
• Reading: ch 9
References
