Clustering and EM

Ron Parr
CPS 271

material from: Lise Getoor, Andrew Moore, Tom Dietterich, Sebastian Thrun, Rich Maclin

Unsupervised Learning

• Supervised learning: Data <x, y>
• Unsupervised Learning: Data x

• So, what’s the big deal?
• Isn’t y just another feature?
• No explicit performance objective
  – Bad news: Problem not necessarily well defined without further assumptions
  – Good news: Results can be useful for more than predicting y
Model Learning

- Produce a global summary of the data
- Not an exact copy
- Assume data are sampled from a larger set that has some easily summarized properties
  - cluster analysis
  - density estimation

- In this lecture: We focus on clustering and use clustering to motivate the EM algorithm

Cluster Analysis

- Decomposition or partition of data into groups where
  - the points in one group are similar to each other
  - and are as different as possible from the points in other groups
Examples of Clustering Applications

- **Marketing**: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs.
- **Land use**: Identification of areas of similar land use in an earth observation database.
- **Insurance**: Identifying groups of motor insurance policy holders with similar claim cost.
- **City-planning**: Identifying groups of houses according to their house type, value, and geographical location.
- **Earth-quake studies**: Observed earth quake epicenters should be clustered along continent faults.

Example

- **Households**: location, income, number of children, rent/own, crime rate, number of cars.

- Appropriate clustering may depend on use:
  - minimize delivery time ⇒ cluster by location
  - others?
  - (Suggests problem is ill defined)
Clustering

• Decomposition or partition of data into groups so that
  – the points in one group are similar to each other
  – and are as different as possible from the points in other groups
• Measure of distance is fundamental
• Explicit representation:
  – $D(x(i), x(j))$ for each $x$
  – only feasible for small domains
• Measurement:
  – distance computed from features
  – we’ve already seen a number of different ways of doing this

Clustering

• Huge body of work
• (aka unsupervised learning, segmentation, …)
• Major difficulty: Measuring success
• Evaluation depends on goals
• If goal is to find ‘interesting’ clusters, this is rather difficult to quantify
• However, for our probabilistic methods, we will present some tools for validating our models
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
• **Graph-based Methods**
  – e.g., spectral methods

Partition-based Clustering Algorithms

• Given set of n data points \(D=\{x^{(1)}, ..., x^{(n)}\}\)
  partition data into k clusters \(C = \{C_1, ..., C_k\}\)
  such that each \(x(i)\) is assigned to a unique \(C_j\)
  and \(\text{Score}(C,D)\) is minimized/maximized
• Combinatorial optimization: searching for allocation of n
  objects into k classes that maximizes score function
• Number of possible allocations = \(k^n\)
• Exhaustive search is intractable
• Resort to iterative improvement
**Possible Scoring Functions**

- **Score function:**
  - clusters compact $\Rightarrow$ minimize within cluster distance, $wc(C)$
  - clusters should be far apart $\Rightarrow$ maximize distance between clusters, $bc(C)$
- Given a clustering $C$, assign cluster centers, $c_k$
  - if points belong to space where means make sense, we can use the centroid of the points in the cluster:
  \[ c_k = \frac{1}{n_k} \sum_{x \in C_k} x \]
- $wc(C) =$ sum-of-squares within cluster distance
  \[ wc(C) = \sum_{k=1}^{K} wc(C_k) = \sum_{k=1}^{K} \sum_{x \in C_k} d(x, c_k) \]
- $bc(C) =$ distance between clusters
  \[ bc(C) = \sum_{1 \leq j < k \leq K} d(c_j, c_k) \]
- Score($C, D$) = $f(wc(C), bc(C))$

---

**K-means**

- Start with randomly chosen cluster centers
- Assign points to closest cluster
- Recompute cluster centers
- Reassign points
- Repeat until no changes
K-means example

X(1)  X(2)  X(3)

X(4)  X(5)  X(6)

X(7)  X(8)

K-means example

X(1)  X(2)  X(3)

X(4)  X(5)  X(6)

X(7)  X(8)

C_1

C_2

C_3
K-means example
K-means example

X(1)  X(2)  X(3)  X(4)  X(5)  X(6)  X(7)  X(8)

C1  C2  C3
K-means example #2

Complexity

- Does algorithm terminate?
- Does algorithm converge to optimal?
- Time complexity one iteration? nk
Understanding k-Means

- Models data as coming from spherical Gaussians centered at cluster centers
- \( \log P(\text{data}) \sim \text{sum of squared distances} \)
- Each step of k-Means increases \( \log P(\text{data}) \)
  - Reassigning
  - Recomputing means

- Fixed number of assignments and monotonic score implies convergence
- (We will later learn that this is an example of EM)

Algorithm Variations

- Recompute centroid as soon as a point is reassigned
- Allow merge and split of clusters
- Cases where means do not make sense
  - \( k \)-mediods – use one of the data points as center
  - Categorical data
- What if data set is too large for algorithm to be tractable?
  - Compress data by replacing groups of objects by ‘condensed representation’
Probabilistic Model-based Clustering

- Assume probability model for each component cluster
- Mixture Model:
  \[ p(x) = \sum_{k=1}^{K} w_k f_k(x; \theta_k) \]
  - where \( f_k \) are component distributions
  - Components: gaussian, poisson, exponential, etc.
  - Most common: Gaussian mixture model (GMM)

Gaussian Mixture Models (GMM)
- \( K \) components
- Model for each component cluster \( f = N(\mu_k, \sigma_k) \)

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

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

\[
p(x) = \sum_{k=1}^{K} p(z_k)p(x | z_k; \mu_k, \sigma_k)
\]

\[
p(x) = \sum_{i=1}^{m} \sum_{k=1}^{K} p(z_k^{(i)}) p(x^{(i)} | z_k^{(i)}; \mu_k, \sigma_k)
\]

- Problem:
  - Non-linear system of equations
  - No efficient analytic solution
  - One solution: gradient descent.... Slow, method of last resort
  - Instead....
Silly Example

Let events be "grades in a class"

- \( w_1 = \) Gets an A \( \quad \) \( \text{P}(A) = \frac{1}{2} \)
- \( w_2 = \) Gets a B \( \quad \) \( \text{P}(B) = \mu \)
- \( w_3 = \) Gets a C \( \quad \) \( \text{P}(C) = 2\mu \)
- \( w_4 = \) Gets a D \( \quad \) \( \text{P}(D) = \frac{1}{2} - 3\mu \)

(Note \( 0 \leq \mu \leq 1/6 \))

Assume we want to estimate \( \mu \) from data. In a given class there were

- a A's
- b B's
- c C's
- d D's

What's the maximum likelihood estimate of \( \mu \) given a,b,c,d ?

---

Trivial Statistics

\( \text{P}(A) = \frac{1}{2} \quad \text{P}(B) = \mu \quad \text{P}(C) = 2\mu \quad \text{P}(D) = \frac{1}{2} - 3\mu \)

\( \text{P}(a,b,c,d \mid \mu) = K(\frac{1}{2})^a(\mu)^b(2\mu)^c(\frac{1}{2} - 3\mu)^d \)

\( \log P(a,b,c,d \mid \mu) = \log K + a \log \frac{1}{2} + b \log \mu + c \log 2\mu + d \log (\frac{1}{2} - 3\mu) \)

For MLE \( \mu \), set \( \frac{\partial \log P}{\partial \mu} = 0 \)

\[ \frac{\partial \log P}{\partial \mu} = \frac{b}{\mu} + \frac{2c}{2\mu} - \frac{3d}{1/2 - 3\mu} = 0 \]

\[ \mu = \frac{b + c}{6(b + c + d)} \]

So if class got

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>14</td>
<td></td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

MLE \( \mu = \frac{1}{10} \)
Same Problem with Hidden Information

Someone tells us that
Number of High grades (A’s + B’s) = \( h \)
Number of C’s = \( c \)
Number of D’s = \( d \)

What is the max. like estimate of \( \mu \) now?

<table>
<thead>
<tr>
<th>REMEMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P(A) = \frac{1}{2} )</td>
</tr>
<tr>
<td>( P(B) = \mu )</td>
</tr>
<tr>
<td>( P(C) = 2\mu )</td>
</tr>
<tr>
<td>( P(D) = \frac{1}{2} - 3\mu )</td>
</tr>
</tbody>
</table>

We can answer this question circularly:

**EXPECTATION**
If we know the value of \( \mu \) we could compute the expected value of \( a \) and \( b \).

Since the ratio \( a:b \) should be the same as the ratio \( \frac{1}{2} : \mu \)

\[
a = \frac{\frac{1}{2}}{1 + \mu} h \\
b = \frac{\mu}{1 + \mu} h
\]

**MAXIMIZATION**
If we know the expected values of \( a \) and \( b \) we could compute the maximum likelihood value of \( \mu \)

\[
\mu = \frac{b + c}{6(b + c + d)}
\]
E.M. for our Trivial Problem

We begin with a guess for \( \mu \)
We iterate between EXPECTATION and MAXIMALIZATION to improve our estimates of \( \mu \) and \( a \) and \( b \).

Define \( \mu(t) \) the estimate of \( \mu \) on the \( t \)th iteration
\( b(t) \) the estimate of \( b \) on \( t \)th iteration

\[
\mu(0) = \text{initial guess} \\
b(t) = \frac{b(t) + \mu(t)}{2 + \mu(t)} = E[b | \mu(t)] \\
\mu(t + 1) = \frac{b(t) + c}{6(b(t) + c + d)} \\
= \text{max like est of } \mu \text{ given } b(t)
\]

Continue iterating until converged.
Good news: Converging to local optimum is assured.
Bad news: "local" optimum.

---

E.M. Convergence

- Convergence proof based on fact that \( \text{Prob(data | } \mu) \) must increase or remain same between each iteration NOT OBVIOUS
- But it can never exceed 1 previous
- So it must therefore converge previous

\[
\begin{array}{c|c|c}
  \text{t} & \mu(t) & b(t) \\
  \hline
  0 & 0 & 0 \\
  1 & 0.0833 & 2.857 \\
  2 & 0.0937 & 3.158 \\
  3 & 0.0947 & 3.185 \\
  4 & 0.0948 & 3.187 \\
  5 & 0.0948 & 3.187 \\
  6 & 0.0948 & 3.187 \\
\end{array}
\]

In our example, suppose we had
\( h = 20 \)
\( c = 10 \)
\( d = 10 \)
\( \mu(0) = 0 \)

Convergence is generally linear; error decreases by a constant factor each time step.
Gaussian Mixture Example: Start

After first iteration
After 2nd iteration

After 3rd iteration
After 4th iteration

After 5th iteration
After 6th iteration

After 20th iteration
Formal EM setup

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

Digression: Jensen’s Inequality

- For convex $f(x)$, density $p(x)$:
  \[
  f \left( \int x p(x) \, dx \right) \leq \int f(x) p(x) \, dx
  \]
- Reversed for concave $f(x)$:
  \[
  f \left( \int x p(x) \, dx \right) \geq \int f(x) p(x) \, dx
  \]
EM Algorithm

- EM algorithm alternates between
  - maximize $F$ with respect to dist. $Q$ with $\theta$ fixed
  - maximize $F$ with respect to $\theta$ with $Q = p(Z)$ fixed

E-step: $Q^{k+1} = \arg \max \limits_{Q} F(Q^k, \theta^k)$

M-step: $\theta^{k+1} = \arg \max \limits_{\theta} F(Q^{k+1}, \theta)$

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

Notes

- Often both E, M steps can be solved in closed form
- Neither E nor M can decrease the log-likelihood
- Convergence guaranteed since $F(Q, \theta)$ nondecreasing
- May converge to local optimum
- Optimum may depend upon initial choice of $\theta, Q$
- Must specify stopping criterion

- Computational complexity depends upon:
  - Number of iterations
  - Time to compute E and M steps
EM Comments

- Complexity of EM for multivariate Gaussian mixtures with K components: dominated by calculation of K covariance matrices.
  - p dimensions: $O(Kp^2)$ 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

and finally...

how do we choose K?
How to choose K

• Choose K that maximizes likelihood?
• No!!!
• As K is increased, the value of the likelihood at maximum cannot decrease
• Problem of scoring models with different complexities
  – Model too flexible ⇒ overfit the data ⇒ high variance
  – Model too restrictive ⇒ can’t fit the data ⇒ high bias
  – Bias-variance tradeoff: compromise
• Solutions:
  – External validation (use k-fold cross validation, LOOCV)
  – Scoring function – MDL, BIC, AIC (complexity penalties)
  – Bayesian model selection

Relationship to K-Means

• Can view K-Means as a special case of EM for GMMs
• Assumes uniform, spherical covariance matrices
• Only hidden parameters are mixture memberships
• Makes “hard” assignments to clusters
  (all probabilities assumed to be 0/1)

• Can show that hard EM converges under similar (minimal) assumptions to those needed for soft EM
Conclusion

- Clustering can be viewed as a problem of inferring hidden parameters of data + model parameters

- EM = powerful and general method for doing this
- Note that EM has applications beyond clustering (can be used any time we need to estimate hidden variables)