Clustering and the EM Algorithm

Susanna Ricco

CPS 271
25 October 2007

Material borrowed from:
Lise Getoor, Andrew Moore, Tom Dietterich, Sebastian Thrun, Rich Maclin, ...
(and Ron Parr)

Unsupervised Learning

Supervised Learning

Given data in the form $< x, y >$, y is the target to learn.

- Good news: Easy to tell if our algorithm is giving the right answer.

Unsupervised Learning

Given data in the form $< x >$ without any explicit target.

- Bad news: How do we define "good performance"?
- Good news: We can use our results for more than just predicting y.

Unsupervised Learning is Model Learning

Goal

Produce global summary of the data.

How?

Assume data are sampled from underlying model with easily summarized properties.

Why?

- Filter out noise
- Data compression

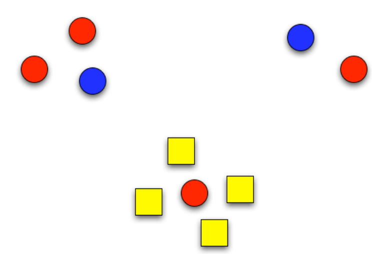
Good Clusters

Want points in a cluster to be:

1. as *similar* as possible to other points in same cluster
2. as *different* as possible from points in another cluster

Warning:
Definition of *similar* and *different* depend on specific application.

We've already seen a lot of ways to measure distance between two data points.



Types of Clustering Algorithms

Hierarchical methods
e.g., hierarchical agglomerative clustering

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

Probabilistic model-based methods
e.g., learning mixture models

Spectral methods
I'm not going to talk about these

Hierarchical Clustering

Build a hierarchy of nested clusters.

Either gradually

- Merge similar clusters (agglomerative method)
- Divide loose superclusters (divisive method)

Result displayed as a *dendrogram* showing the sequence of merges or splits.

Agglomerative Hierarchical Clustering

Initialize $C_i = \{x^{(i)}\}$ for $i \in [1, n]$.

While more than one cluster left:

1. Let C_i, C_j be clusters that minimize $D(C_i, C_j)$
2. $C_i = C_i + C_j$
3. Remove C_j from list of clusters
4. Store current clusters

Measuring Distance

What is $D(C_i, C_j)$?

- Single link method:
$$D(C_i, C_j) = \min\{d(x, y) | x \in C_i, y \in C_j\}$$
- Complete link method:
$$D(C_i, C_j) = \max\{d(x, y) | x \in C_i, y \in C_j\}$$
- Average link method:
$$D(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y)$$
- Centroid measure:
$$D(C_i, C_j) = d(c_i, c_j), \text{ where } c_i \text{ and } c_j \text{ are centroids}$$
- Ward's measure:
$$D(C_i, C_j) = \sum_{x \in C_i} d(x, \bar{x}) + \sum_{y \in C_j} d(y, \bar{y}) - \sum_{u \in C_i \cup C_j} d(u, \bar{u})$$

Result

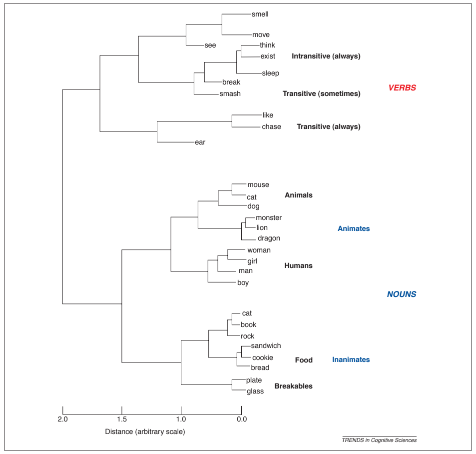


Figure 2. Hierarchical clustering diagram of hidden-unit activation patterns in response to different words. The similarity between words and groups of words is reflected in the tree structure; items that are closer are joined further down the tree (i.e. to the right as shown here).

Elman, J.L. An alternative view of the mental lexicon. *Trends in Cognitive Science*, 7, 301-306.

Divisive Hierarchical Clustering

Begin with one single cluster, split to form smaller clusters.

Can be difficult to choose potential splits:

- Monolithic methods split based on values a single variable
- Polythetic methods consider all variables together

Less popular than agglomerative methods.

Partition-based Clustering

Pick some number of clusters K

Assign each point $x^{(i)}$ to a single cluster C_k so that $SCORE(C, D)$ is minimized/maximized.

- (What is the score function?)

Total number of possible allocations: k^n

Use iterative improvement instead of intractable exhaustive search.

The K-Means Algorithm

A popular partition-based clustering algorithm with the score function given by:

$$SCORE(C, D) = \sum_{k=1}^{K} d(x, c_k)$$

where

$$c_k = \frac{1}{n_k} \sum_{x \in C_k} x$$

and

$$d(x, y) = ||x - y||^2.$$

Pseudo-code for K-Means

1. Initialize k cluster centers, c_k.
2. For each $x^{(i)}$, assign cluster with closest center

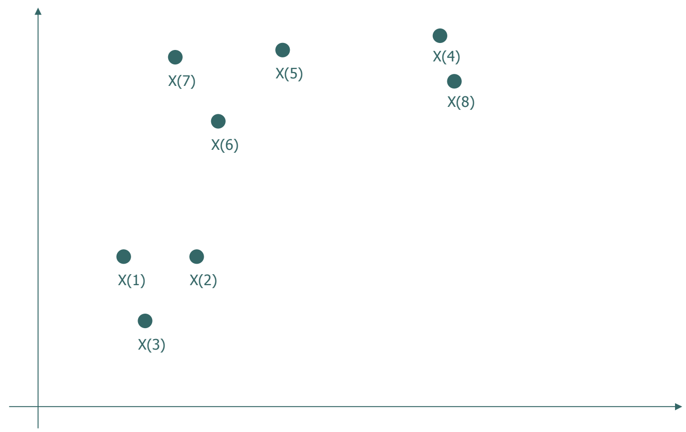
$$x^{(i)} \text{ assigned to } \hat{k} = \arg\min_k d(x, c_k).$$

3. For each cluster, recompute center:

$$c_k = \frac{1}{n_k} \sum_{x \in C_k} x$$

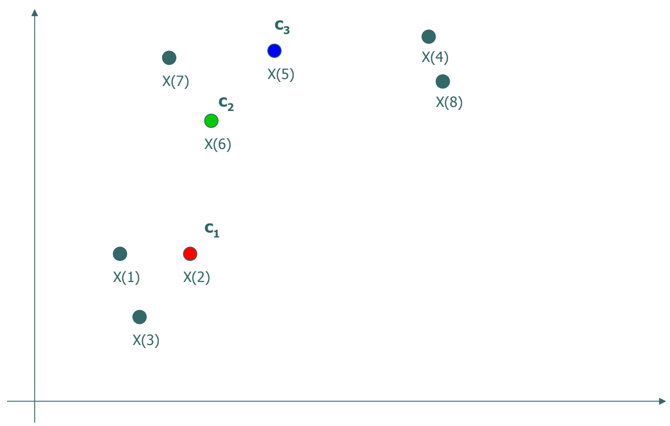
4. Check convergence (Have cluster centers moved?)
5. If not converged, go to 2.

K-Means Example



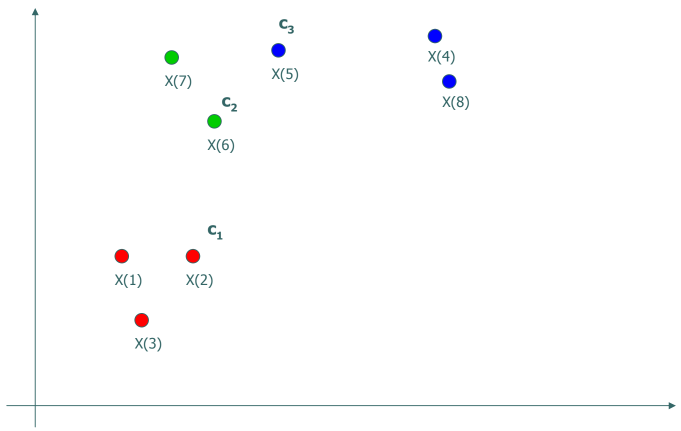
Original unlabeled data.

K-Means Example



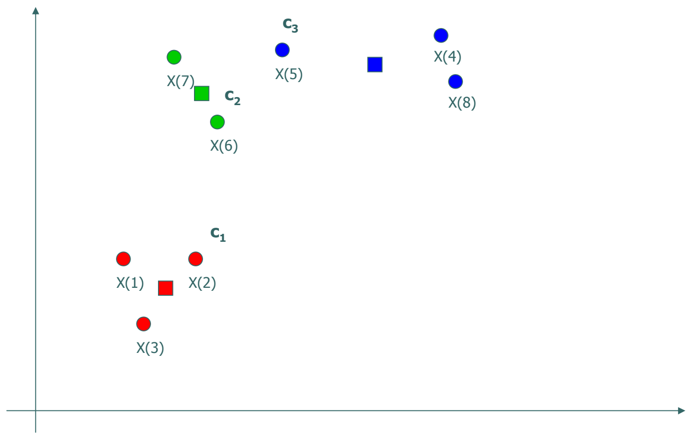
Pick initial centers randomly.

K-Means Example



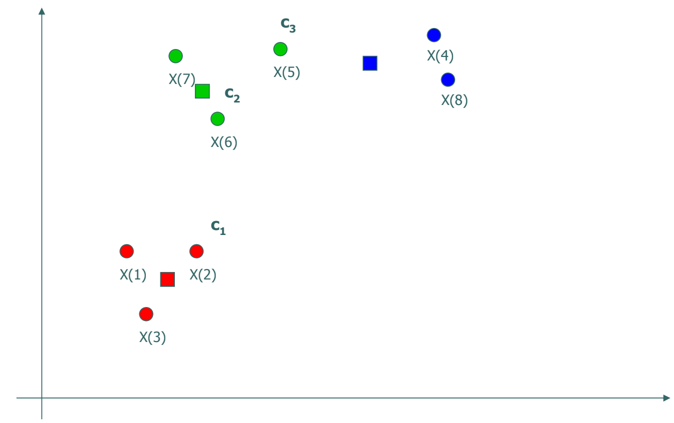
Assign points to nearest cluster.

K-Means Example



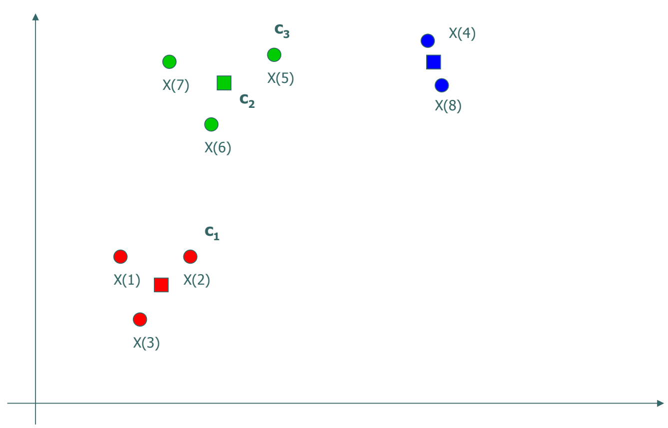
Recompute cluster centers.

K-Means Example



Reassign points to nearest clusters.

K-Means Example



Recompute cluster centers.

Understanding K-Means

Time complexity per iteration?

Does algorithm terminate?

Does algorithm converge to global optimum?

K-Means Convergence

Model data as drawn from spherical Gaussians centered at cluster centers.

$$\log P(data|assignments) = const - \frac{1}{2}\sum_{k=1}^{K}\sum_{x \in C_k}(x - c_k)^2.$$

- How does this change when we reassign a point?
- How does this change when we recompute the means?

Monotonic improvement + finite assignments = convergence.

Demo

`http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html`

Variations on K-Means

What if we don't know K?
Allow merging or splitting of clusters using heuristics.

What if means don't make sense?
Use k-*mediods* instead.

Mixture Models

Assume data generated using the following procedure.

1. Pick one of k components according to $P(z_k)$.
   This selects a (hidden) class label z_k.
2. Generate a data point by sampling from $p(x|z_k)$.

Results in probability distribution of single point

$$p(x^{(i)}) = \sum_{k=1}^{K} P(z_k)p(x^{(i)}|z_k)$$

where $p(x|z_k)$ is any distribution (gaussian, poisson, exponential, etc.).

Gaussian Mixture Model (GMM)

Most common mixture model is a Gaussian mixture model:

$$p(x|z_k) = \mathcal{N}(\mu_k, \Sigma_k)$$

With this model, likelihood of data becomes

$$p(x) = \sum_{n=1}^{N}\sum_{k=1}^{K} P(z_k)p(x^{(i)}|z_k; \mu_k, \Sigma_k).$$

LDA and GMMs

LDA

- Built models $p(x|z_k)$ and $P(z_k)$ using maximum likelihood given our training data.
- Used these models to compute $P(z_k|x)$ to classify new query points.

Clustering with GMMs

- Want to find $P(z_k)$ and $p(x|z_k)$ to learn underlying model and find clusters.
- Want to compute $P(z_k|x)$ for each point in training set to assign them to clusters.
- Can we use maximum likelihood to infer both model and assignments?
  - Requires solving non-linear system of equations
  - No efficient analytic solution

Problem: Missing Labels

If we knew assignments, we could learn component models easily.

- We did this to train an LDA.

If we new the component models, we could estimate the most likely assignments easily.

- This is just classification.

Solution: The Expectation Maximization (EM) Algorithm

We deal with missing labels by alternating between two steps:

1. Expectation: Fix model and estimate missing labels.
2. Maximization: Fix missing labels (or a distribution over the missing labels) and find the model that maximizes the expected log-likelihood of the data.

Simple Example

Labeled Data

Clusters correspond to "grades in class".

Model to learn:

$$
\begin{aligned}
P(A) &= \frac{1}{2} \\
P(B) &= \mu \\
P(C) &= 2\mu \\
P(D) &= \frac{1}{2} - 3\mu
\end{aligned}
$$

Training data:

a people got an A

b people got a B

c people got a C

d people got a D

What is maximum likelihood estimate for μ?

Simple Example

Labeled Data

Likelihood:

$$
\begin{aligned}
P(a,b,c,d|\mu) &= K\left(\frac{1}{2}\right)^a (\mu)^b (2\mu)^c \left(\frac{1}{2} - 3\mu\right)^d \\
\log P(a,b,c,d|\mu) &= \log K + a\log\frac{1}{2} + b\log\mu + c\log(2\mu) + d\log\left(\frac{1}{2} - 3\mu\right) \\
\frac{\partial}{\partial\mu}\log P(a,b,c,d|\mu) &= \frac{b}{\mu} + \frac{2c}{2\mu} - \frac{3d}{\frac{1}{2} - 3\mu}
\end{aligned}
$$

For MLE, set $\frac{\partial}{\partial\mu}\log P = 0$ and solve for μ to get

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

Simple Example

Hidden Labels

What if we only know that there are h "high grades"? (Exact labels are missing.)

Now how do we find the maximum likelihood estimate of μ?

1. Expectation:
   Fix μ and infer the expected values of a and b:

$$a = \frac{1/2}{1/2+\mu}h, \quad b = \frac{\mu}{1/2+\mu}h$$

   Since we know $\frac{a}{b} = \frac{1/2}{\mu}$ and $a + b = h$.
2. Maximization:
   Fix these fractions a and b and compute the maximum likelihood μ as before:

$$\mu_{new} = \frac{b+c}{6(b+c+d)}.$$

3. Repeat.

Formal Setup for General EM Algorithm

Let $D = \{x^{(1)}, \ldots, x^{(n)}\}$ be n observed data vectors.

Let $Z = \{z^{(1)}, \ldots, z^{(n)}\}$ be n values of hidden variables (i.e., the cluster labels).

Log-likelihood of observed data given model:

$$L(\theta) = \log p(D|\theta) = \log \sum_Z p(D, Z|\theta)$$

Note: both θ and Z are unknown.

Fun with Jensen's Inequality

Let $Q(Z)$ be any distribution over the hidden variables:

$$\begin{aligned}
\log P(D|\theta) &= \log \sum_Z Q(Z) \frac{p(D, Z|\theta)}{Q(Z)} \\
&\geq \sum_Z Q(Z) \log \frac{p(D, Z|\theta)}{Q(Z)} \\
&= \sum_Z Q(Z) \log p(D, Z|\theta) + \sum_Z Q(Z) \log \frac{1}{Q(Z)} \\
&= F(Q, \theta)
\end{aligned}$$

General EM Algorithm

Alternate between steps until convergence:

E step:

- Maximize F wrt Q, keeping θ fixed.
- Solution:

$$Q^{k+1} = p(Z|D, \theta^k)$$

M step:

- Maximize F wrt θ, keeping Q fixed
- Solution:

$$\begin{aligned}
\theta^{k+1} &= \arg\max_\theta F(Q^{k+1}, \theta) \\
&= \arg\max_\theta \sum_Z p(Z|D, \theta^k) \log p(X, Z|\theta)
\end{aligned}$$

General EM Algorithm in English

Alternate steps until model parameters don't change much:

E step:

Estimate distribution over labels given a certain fixed model.

M step:

Choose new parameters for model to maximize expected log-likelihood of observed data and hidden variables.

Convergence

The EM Algorithm will converge because:

- During E step, we make $F(Q^{k+1}, \theta^k) = \log P(D|\theta^k)$.
- During M step, we choose θ^{k+1} that increases F.
- Recall that F is a lower bound,

$$F(Q^{k+1}, theta^{k+1}) \leq \log P(D|\theta^{k+1}).$$

- Implies

$$\log P(D|\theta^k) \leq \log P(D|\theta^{k+1})$$

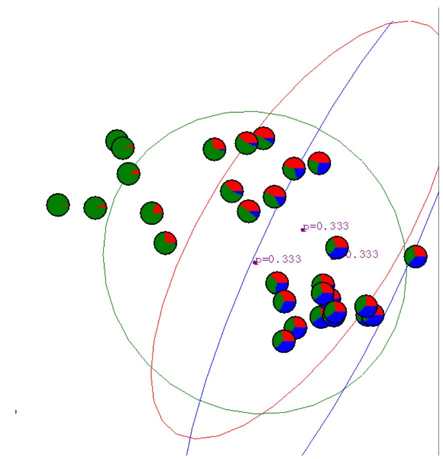
- Implies convergence! (Why?)

Notes

Things to remember:

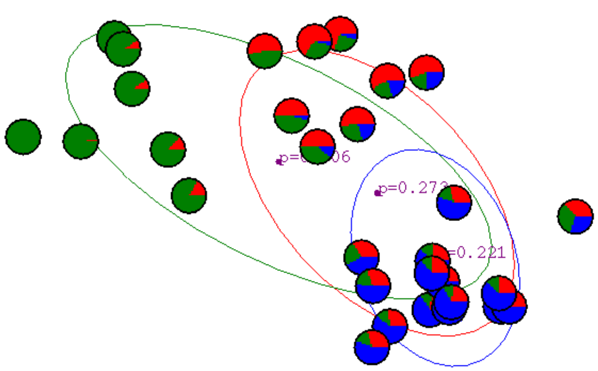
- Often closed form for both E and M step.
- Must specify stopping criteria.
- Complexity depends on number of iterations and time to compute E and M steps.
- May (will) converge to local optimum.

Example: EM for GMM



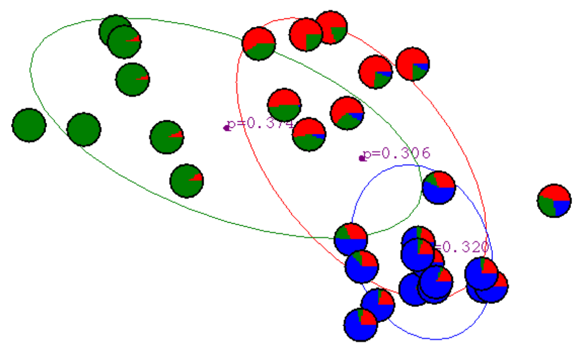
Initial model parameters.

Example: EM for GMM



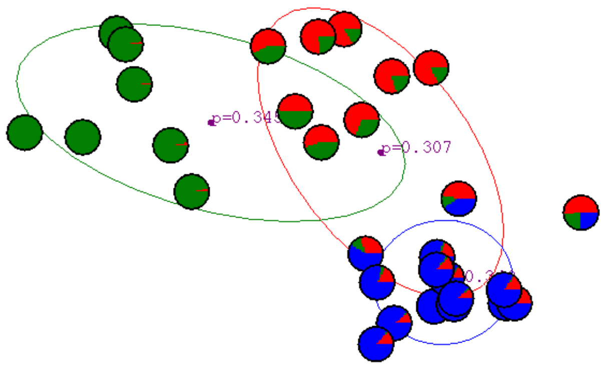
After first iteration

Example: EM for GMM



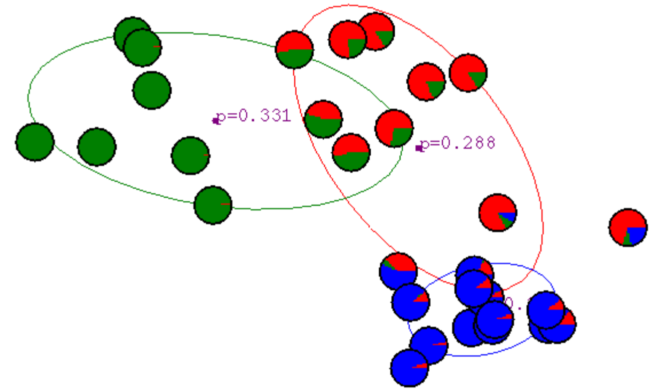
After second iteration

Example: EM for GMM



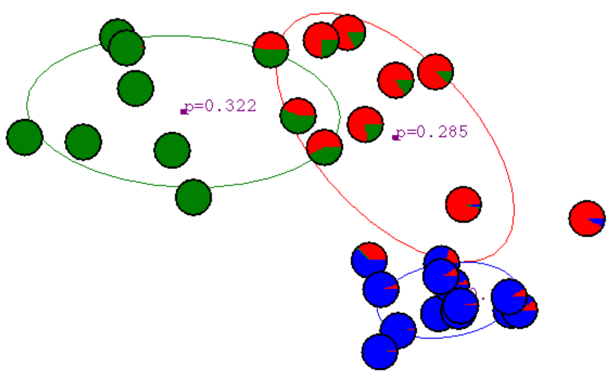
After third iteration

Example: EM for GMM



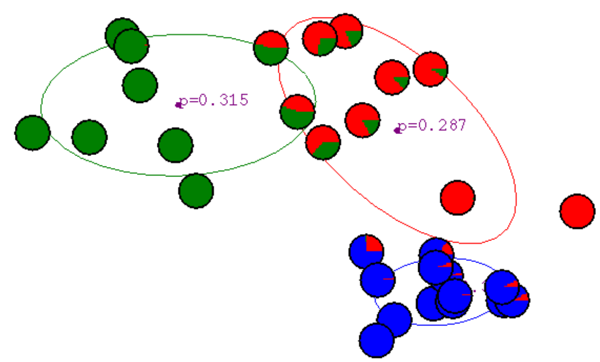
After fourth iteration

Example: EM for GMM



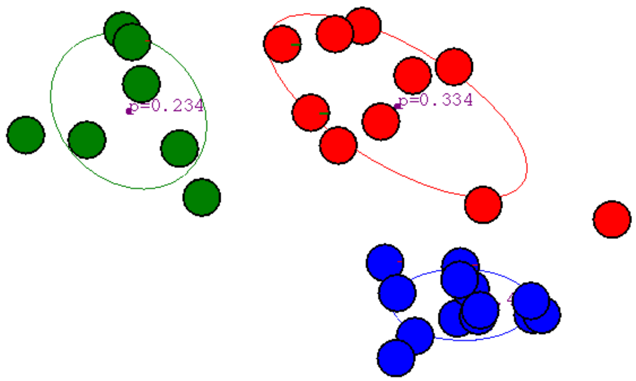
After fifth iteration

Example: EM for GMM



After sixth iteration

Example: EM for GMM



After convergence

Relation to K-Means

Similarities

K-Means used GMM with:

- covariance $\Sigma = I$ (fixed)
- uniform $P(Z_k)$ (fixed)
- unknown means

Alternated estimating labels and recomputing unknown model parameters.

Difference

Makes "hard" assignment to cluster during E step.

How to Pick K?

Do we want to pick the K that maximizes likelihood?

How to Pick K?

Do we want to pick the K that maximizes likelihood?

Other options:

- Cross-validation
- Add complexity penalty to objective function
- Prior knowledge

Summary

Clustering:

Infer assignments to hidden variables and hidden model parameters simultaneously.

EM Algorithm:

Powerful, popular, general method for doing this.

EM Applications:

- Image segmentation
- SLAM
- Estimating motion models for tracking
- Hidden Markov Models
- etc.