9.913 Pattern Recognition for Vision

Class VII, Part I – Techniques for Clustering

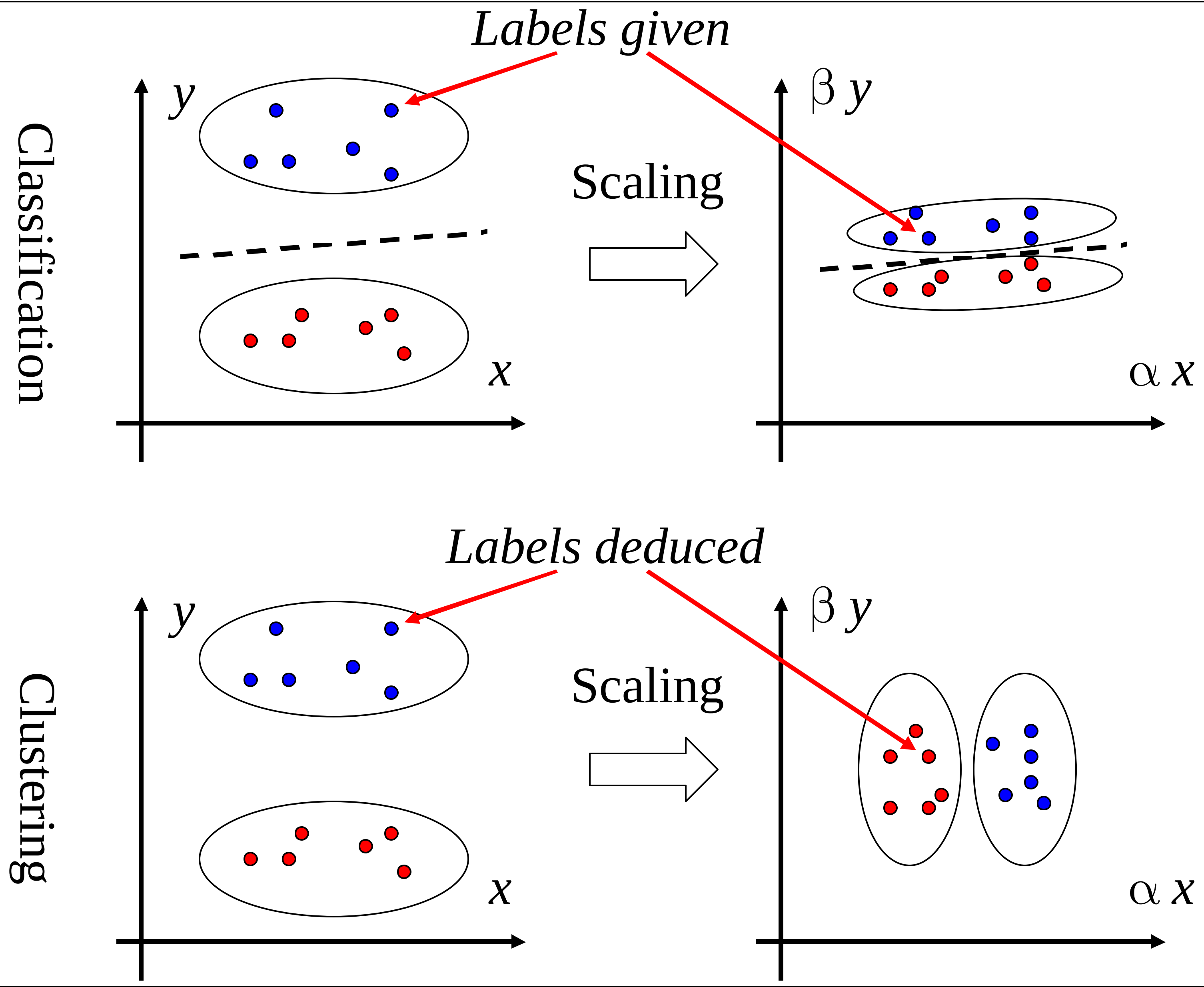
Yuri Ivanov

TOC

Clustering

- Clustering is a process of partitioning the data into groups based on similarity
- Clusters are groups of measurements that are *similar*
- In *Classification* groups of similar data form classes
  - Labels are given
  - Similarity is deduced from labels
- In Clustering groups of similar data form clusters
  - Similarity measure is given
  - Labels are deduced from similarity

Clustering

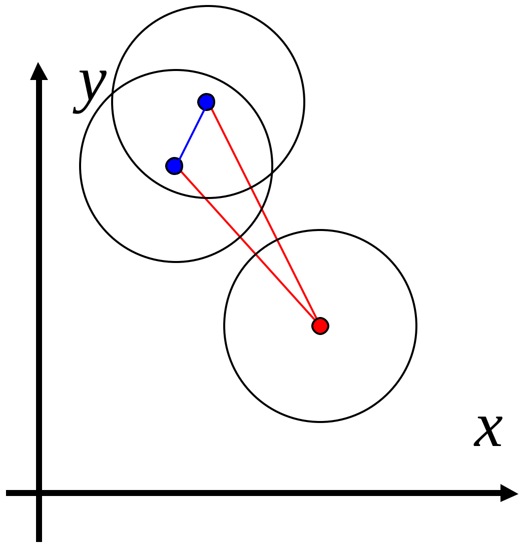

Labels given
Classification
y
x
Scaling
β y
α x
Labels deduced
Clustering
y
x
Scaling
β y
α x

Questions

- What is “similar”?
- What is a “good” partitioning?

Distances

Most obvious: distance between samples
- Compute distances between the samples
- Compare distances to a threshold



We need a metric to define distances and thresholds

Metric and Invariance

- We can choose it from a family:

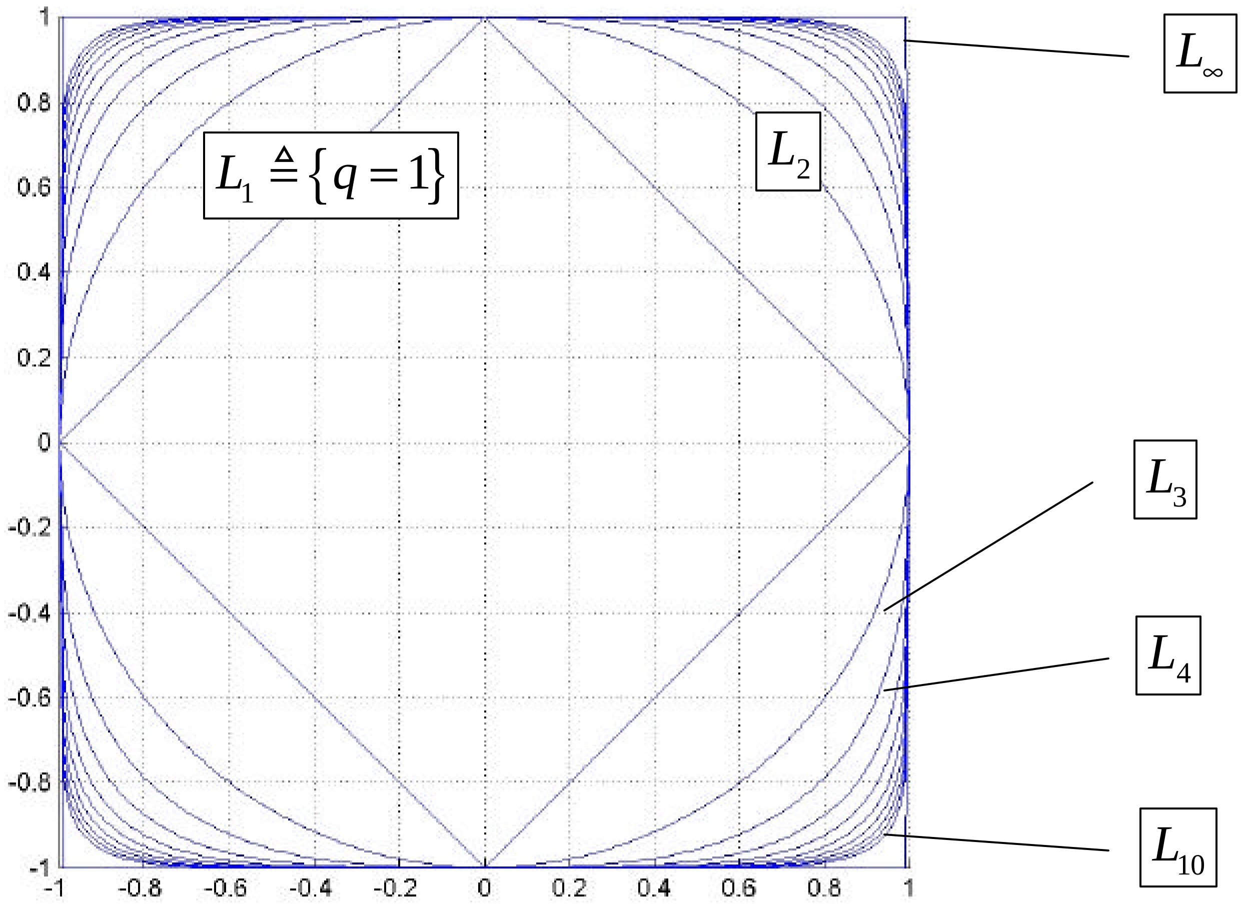
$$d(x, x') = \left(\sum_{k=1}^{d} \left| x_k - x'_k \right|^q \right)^{1/q}$$ - Minkowski metric

$q = 1 \;\Rightarrow$ Manhattan/city block/taxicab distance

$q = 2 \;\Rightarrow$ Euclidean distance

$d(x, x')$ is invariant to rotation and translation only for $q = 2$

Minkovski Metrics



Points a distance 1 from origin

Metric and Invariance

Other choices for invariant metric:

- We can use data-driven metric:

$$d(x, x') = \sqrt{(x - x')^T \Sigma^{-1} (x - x')}$$ - Mahalanobis distance

- We can normalize data (whiten)

$$x' = \left(\Lambda^{-1/2} \Phi^T\right) x$$

And then use the Euclidean metric

Metric

Euclidean metric
- Good for isotropic spaces
- Bad for linear transformations (except rotation and translation)

Mahalanobis metric:
- Good if there is enough data

Whitening:
- Good if the spread is due to random processes
- Bad if it is due to subclasses

Similarity

We need a symmetric function that is large for "similar" *x*

E.g.: $$s(x, x') = \frac{x^t x'}{\|x^t\| \|x'\|}$$ - "angular" similarity

Vocabulary:

{*Two, three, little, star, monkeys, jumping, twinke, bed* }

a) *Three little monkeys jumping on the bed*	(0, 1, 1, 0, 1, 1, 0, 1)
b) *Two little monkeys jumping on the bed*	(1, 0, 1, 0, 1, 1, 0, 1)
c) *Twinkle twinkle little star*	(0, 0, 1, 1, 0, 0, 2, 0)

Similarity matrix:

	a	b	c
a	1.0	0.8	0.18
b	0.8	1.0	0.18
c	0.18	0.18	1.0

Similarity

It doesn’t have to be metric:

E.g.:

	Has fur	Has 4 legs	Can type
Monkey	1	0	1
Platypus	1	1	0

$$s(x, x') = \frac{x^t x'}{d}$$

.67	.33
.33	.67

$$s(x, x') = \frac{x^t x'}{\underbrace{x^t x + x'^t x' - x^t x'}_{\textit{Tanimoto coefficient}}}$$

1	.33
.33	1

Partitioning Evaluation

J – objective function, s.t. clustering is assumed optimal when *J* is minimized or maximized

$$J = \sum_{k=1}^{K} \sum_{n=1}^{N_k} \left\| x_n^{(k)} - m_k \right\|^2$$ - Sum of squared error criterion (min)

Using the definition of the mean:

$$J = \frac{1}{2} \sum_{k=1}^{K} N_k \left[\frac{1}{N_k^2} \sum_{n=1}^{N_k} \sum_{m=1}^{N_k} \left\| x_n^{(k)} - x_m^{(k)} \right\|^2 \right]$$

*Dis*similarity measure

You can replace it with your favorite

Partitioning Evaluation

Other possibilities:

For within- and between- cluster scatter matrices (recall LDA)

$$J = \left|S_W\right| = \left|\sum_{k=1}^{K} S_k\right|$$ - Scatter determinant criterion (min)

$$J = tr\left|S_W^{-1} S_B\right| = \sum_{i=1}^{d} l_i$$ - Scatter ratio criterion (max)

Careful with the ranks!

Which to choose?

- No methodological answer

- SSE criterion (minimum variance)
  - simple
  - good for well separated clusters in dense groups
  - affected by outliers, scale variant

- Scatter criteria
  - Invariant to general linear transformations
  - Poor on small amounts of data as related to dimensionality

- You should chose the metric and the objective that are invariant to the transformations natural to your problem

Clustering

$\boldsymbol{x}$ – input data

K – number of clusters (assumed known)

N_k – number points in cluster k

N – total number of data points

$\boldsymbol{t}_k$ – prototype (template) vector of k-th cluster

J – objective function, s.t. clustering is assumed optimal when J is extremized

General Procedure

Clustering is usually an iterative procedure:

- Choose initial configuration
- Adjust configuration s.t. J is optimized
- Check for convergence

J is often only *partially* minimized.

Clustering – A Good Start

Let’s choose the following model:

- Known number of clusters
- Each cluster is represented by a single prototype
- Similarity is defined in the nearest neighbor sense

Sum-Squared-Error objective:

$$J = \sum_{k=1}^{K} \sum_{n=1}^{N_k} \left\| x_n^{(k)} - t_k \right\|^2$$ - total in-cluster distance for all clusters

$$\frac{dJ}{dt_k} = \sum_{c=1}^{K} \sum_{n=1}^{N_k} \frac{d}{dt_k} \left(\left\| x_n^{(k)} - t_k \right\|^2 \right) = -2 \sum_{n=1}^{N_k} (x_n^{(k)} - t_k) = 0 \Rightarrow$$

$$t_k = \frac{1}{N_k} \sum_{n=1}^{N_k} x_n^{(k)}$$

K-Means Algorithm

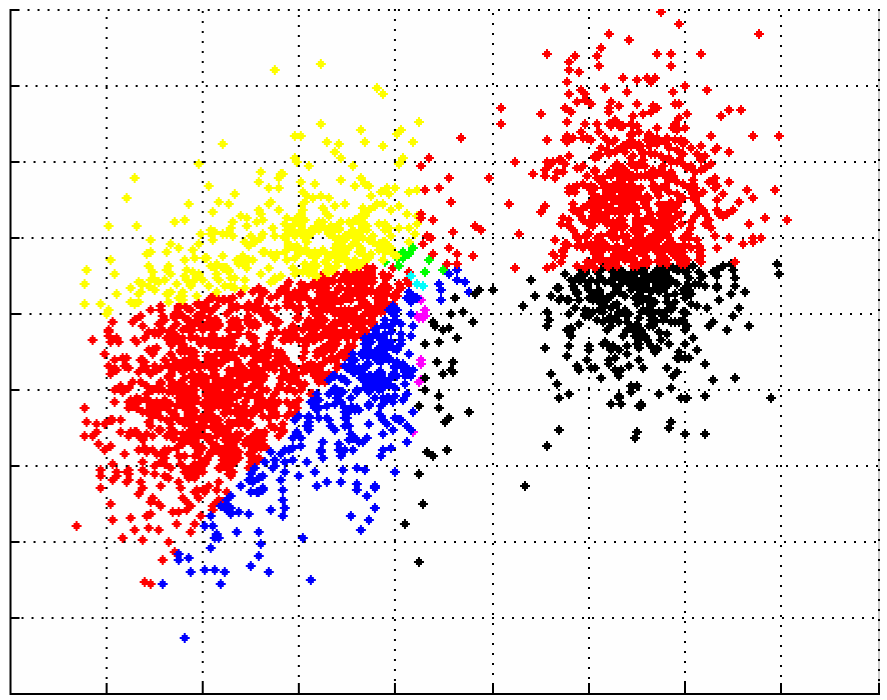
Using the iterative procedure:

1. Choose M random positions for the prototypes
2. Classify all samples by the nearest $\boldsymbol{t}_k$
3. Compute new prototype positions
4. If not converged (no cluster assignments changed from previous iteration), go to step 2

This is the *K-Means* (a.k.a. Lloyd's, a.k.a. LBG) algorithm.

What to do with empty clusters? Some heuristics are involved.

K-Means Algorithm Example



K = *10*

Cluster Heuristics

Sometimes clusters end up empty. We can:
- Remove them
- Randomly reinitialize them
- Split the largest ones

Sometimes we have too many clusters. We can:
- Remove the smallest ones
- Relocate the smallest ones
- Merge the smallest ones together if they are neighbors

IsoData Algorithm

In *K-Means* we assume that we know the number of clusters

IsoData tries to estimate them – ultimate *K-Means* hack

IsoData iterates between 3 stages:

- Center estimation
- Cluster splitting
- Cluster merging

The user specifies:

T – min number of samples in a cluster

N_D – <u>*desired*</u> number of clusters

D_m – max distance for merging

σ_S^2 – maximum cluster variance

N_{max} – max number of merges

IsoData

Stage I – Cluster assignment:

1. Assign a label to each data point such that:

$$w^n = \arg\min_j \left\| x^n - t_j \right\|$$

2. Discard clusters with $N_k < T$, reduce N_c

3. Update means of remaining clusters:

$$t_j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_i^{(j)}$$

This is basically a step of *K-Means* algorithm

IsoData

Stage II – Cluster splitting:

1. If this is the last iteration, set $D_m=0$ and go to Stage III
2. If $N_c<=N_D/2$, go to splitting (step 4)
3. If iteration is even or if $N_c>=2N_D$ go to Stage III
4. Compute:

$$d_k = \frac{1}{N_k}\sum_{i=1}^{N_k}\left\| x_i^{(k)} - t_k \right\|$$ - avg. distance from the center

$$s_k^2 = \max_j \frac{1}{N_k}\sum_{i=1}^{N_k}\left(x_{i,j}^{(k)} - t_{k,j} \right)^2$$ - max variance along a single dimension

$$d = \frac{1}{N}\sum_{k=1}^{N_c} N_k d_k$$ - overall avg. distance from centers

IsoData

Stage II – Cluster splitting (cont.):

5. For clusters with $\sigma_k^2 > \sigma_S^2$:

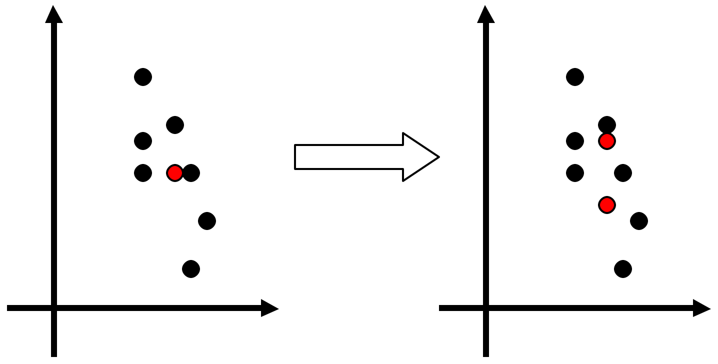
   If ($d_k > d$ AND $N_k > 2(T+1)$) OR $N_c < N_D/2$

   Split the cluster by creating a new mean:

$$t'_{k,j} = t_{k,j} + 0.5 s_k^2$$

   And moving the old one to:

$$t_{k,j} = t_{k,j} - 0.5 s_k^2$$



IsoData

Stage III – Cluster merging:

If no split has been made:

1. Compute the matrix of distances between cluster centers

$$D_{i,j} = \left\| t_i - t_j \right\|$$

2. Make the list of pairs where $D_{i,j} < D_m$

3. Sort them in ascending order

4. Merge up to N_{max} *unique* pairs starting from the top by removing t_j and replacing t_i with:

$$t_i = \frac{1}{N_i + N_j}\left(N_i t_i + N_j t_j \right)$$

IsoData Example

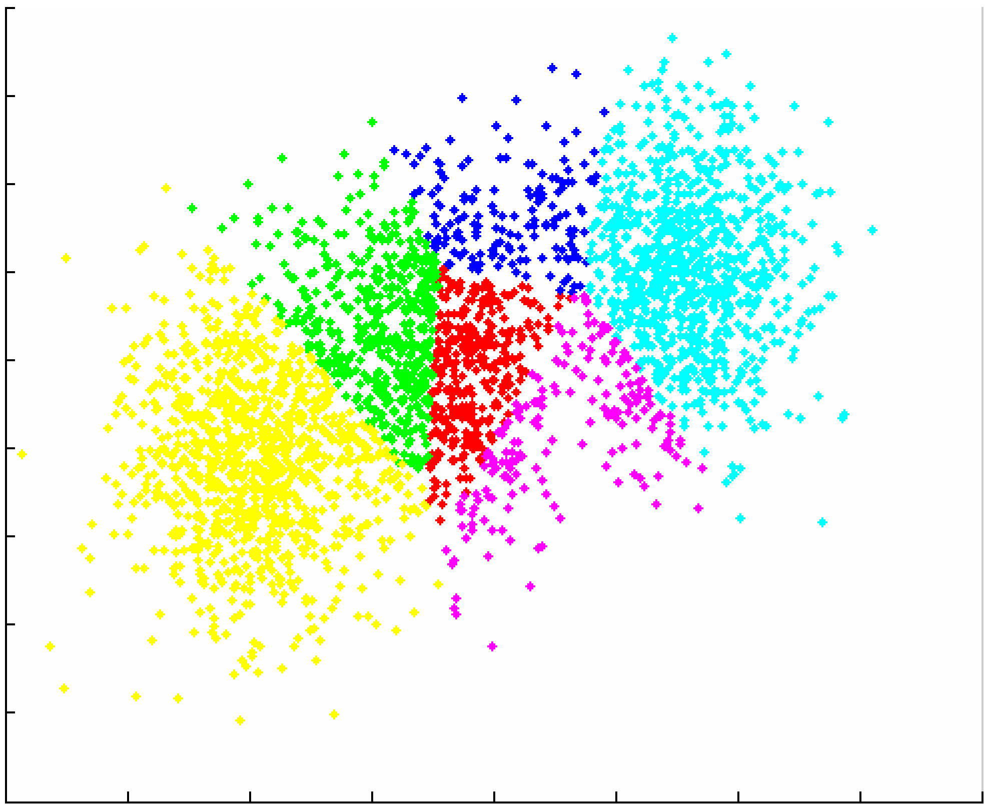
$N_D = 10$

$T = 10$

$\sigma_S^2 = 3$

$D_m = 2$

$N_{max} = 3$



Mixture Density Model

Mixture model – a linear combination of parametric densities

Number of components

$$p(x) = \sum_{j=1}^{M} p(x \mid j) P(j)$$

Component density

Component weight

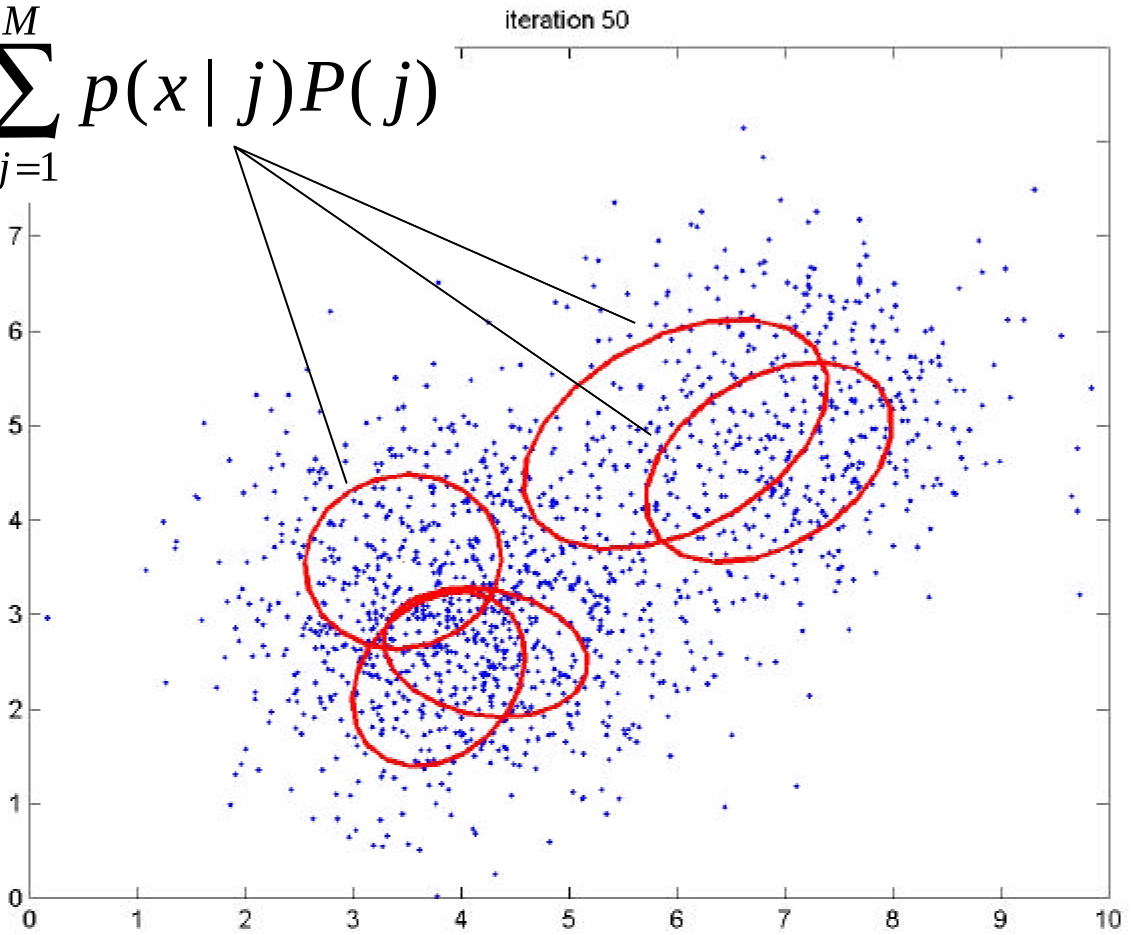
$$P(j) \geq 0, \quad \forall j \qquad \text{and} \qquad \sum_{j=1}^{M} P(j) = 1$$

Recall Kernel density estimation

Kernels are parametric densities, subject to estimation

Example

$$p(x) = \sum_{j=1}^{M} p(x \mid j) P(j)$$



Mixture Density

Using ML principle, the objective function is the *log-likelihood*:

$$l(q) \equiv \ln\left\{\prod_{n=1}^{N} p(x^n)\right\} = \sum_{n=1}^{N} \ln\left\{\sum_{j=1}^{M} p(x^n \mid j)P(j)\right\}$$

Diffirentiate w.r.t. parameters:

$$\nabla_{q_j} l(q) = \sum_{n=1}^{N} \frac{\partial}{\partial \mathbf{q}_j} \ln\left\{\sum_{k=1}^{M} p(x^n \mid k)P(k)\right\}$$

$$= \sum_{n=1}^{N} \boxed{\frac{1}{\sum_{k=1}^{M} p(x^n \mid k)P(k)}} \frac{\partial}{\partial \mathbf{q}_j} p(x^n \mid j)P(j)$$

Here because of the log

Mixture Density

For distributions $p(x|j)$ in the exponential family:

$$\frac{\partial}{\partial q}\left[\boxed{A(q)e^{B(q,x)}}\right] = \boxed{A(q)e^{B(q,x)}}\frac{\partial}{\partial q}\left[B(q,x)\right] + \frac{\partial}{\partial q}\left[A(q)\right]e^{B(q,x)}$$

Goes in here *Goes in here* *Goes in here*

$$\Rightarrow \frac{\partial l(q)}{\partial q} = \sum_{n=1}^{N} P(j\,|\,x^n)\times(Stuff + More\,Stuff)$$

For a Gaussian:

$$\frac{\partial l(q)}{\partial m_j} = \sum_{n=1}^{N} P(j\,|\,x^n)\left[\Sigma_j^{-1}(x^n - \hat{m}_j)\right]$$

$$\frac{\partial l(q)}{\partial \hat{\mathbf{S}}_j} = \sum_{n=1}^{N} P(j\,|\,x^n)\left[\hat{\mathbf{S}}_j^{-1} - \hat{\mathbf{S}}_j^{-1}(x^n - \hat{m}_j)(x^n - \hat{m}_j)^T\,\hat{\mathbf{S}}_j^{-1}\right]$$

Mixture Density

At the extremum of the objective:

$$P(j) = \frac{1}{N}\sum_{n=1}^{N} P(j \mid x^n)$$

$$\hat{m}_j = \frac{\sum_{n=1}^{N} P(j \mid x^n)\, x^n}{\sum_{n=1}^{N} P(j \mid x^n)}$$

$$\hat{\mathbf{S}}_j = \frac{\sum_{n=1}^{N} P(j \mid x^n)\left(x^n - \hat{m}_j\right)\left(x^n - \hat{m}_j\right)^T}{\sum_{n=1}^{N} P(j \mid x^n)}$$

BUT:

$$P(j \mid x^n) = \frac{p(x^n \mid j)\, P(j)}{\sum_{k=1}^{M} p(x^n \mid k) P(k)}$$ - parameters are tied

Solution – EM algorithm.

EM Algorithm

Suppose we pick an initial configuration (just like in K-Means)

Recall the objective (change of sign):

$$E \equiv -l(q) = -\ln\left\{\prod_{n=1}^{N} p(x^n)\right\} = -\sum_{n=1}^{N} \ln\left\{p(x^n)\right\}$$

After a single step of optimization:

$$E^{new} - E^{old} = -\sum_{n=1}^{N} \ln\left\{\frac{p^{new}(x^n)}{p^{old}(x^n)}\right\}$$

$$= -\sum_{n=1}^{N} \ln\left\{\sum_{j=1}^{M} \frac{P^{new}(j)\, p^{new}(x^n \mid j)}{p^{old}(x^n)}\right\}$$

EM Algorithm

After optimization step:

$$E^{new} - E^{old} = -\sum_{n=1}^{N} \ln\left\{\sum_{j=1}^{M} \frac{P^{new}(j)\,p^{new}(x^n \mid j)}{p^{old}(x^n)}\right\}$$

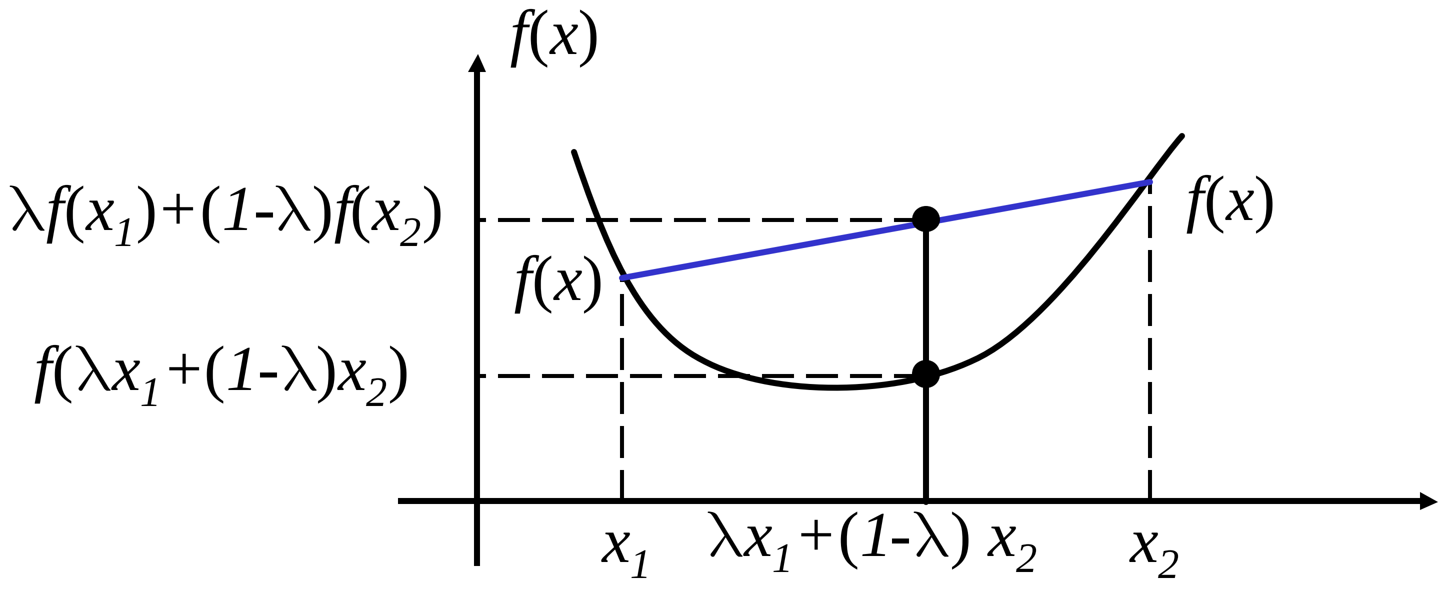
$$= -\sum_{n=1}^{N} \ln\left\{\sum_{j=1}^{M}\left[\frac{P^{new}(j)\,p^{new}(x^n \mid j)}{p^{old}(x^n)} \underbrace{\boxed{\frac{P^{old}(j \mid x^n)}{P^{old}(j \mid x^n)}}}_{= 1}\right]\right\}$$

$$= -\sum_{n=1}^{N} \underbrace{\ln\left\{\sum_{j=1}^{M}\left[\underbrace{\boxed{P^{old}(j \mid x^n)}}_{\text{Sums to 1 over } j} \frac{P^{new}(j)\,p^{new}(x^n \mid j)}{p^{old}(x^n)\,P^{old}(j \mid x^n)}\right]\right\}}_{\boxed{\ln\left\{\sum_{j=1}^{M} l_j y_j\right\}}}$$

Digression-Convexity

Definition: Function f is convex on $[a, b]$ iff for any x_1, x_2 in $[a, b]$ and any λ in $[0, 1]$:

$$f\left(l\, x_1 + (1-l)x_2\right) \leq l\, f\left(x_1\right) + (1-l)\, f\left(x_2\right)$$



$f(x)$
$\lambda f(x_1)+(1-\lambda)f(x_2)$
$f(x)$
$f(x)$
$f(\lambda x_1+(1-\lambda)x_2)$
x_1
$\lambda x_1+(1-\lambda)\, x_2$
x_2

Digression - Jensen's Inequality

If f is a convex function:

$$f\left(\sum_{j=1}^{M} l_j x_j\right) \leq \sum_{j=1}^{M} l_j f(x_j)$$

$$\forall l: \; 0 \leq l_j \leq 1, \qquad \sum_j l_j = 1$$

Equivalently: $$f\left(E[x]\right) \leq E[f(x)]$$

Or: $$f\left(\frac{1}{M}\sum_{j=1}^{M} x_j\right) \leq \frac{1}{M}\sum_{j=1}^{M} f(x_j)$$

Flip the inequality if f is concave

Digression - Jensen's Inequality

Proof by induction:

a) *JE* is trivially true for any 2 points (definition of convexity)

b) Assuming it is true for any *k-1* points:

for $l_i^* \triangleq l_i / (1 - l_k)$

$$\sum_{i=1}^{k} l_i f(x_i) = l_k f(x_k) + (1 - l_k) \sum_{i=1}^{k-1} l_i^* f(x_i)$$

$$\geq l_k f(x_k) + (1 - l_k) f\left(\sum_{i=1}^{k-1} l_i^* x_i \right)$$

$$\geq f\left(l_k x_k + (1 - l_k) \sum_{i=1}^{k-1} l_i^* x_i \right) = f\left(\sum_{i=1}^{k} l_i x_i \right)$$

End of digression

Back to EM

Change in the error:

$$E^{new} - E^{old} =$$

$$= -\sum_{n=1}^{N} \ln \overbrace{\left\{ \sum_{j=1}^{M} \left[\underset{\uparrow \atop \lambda}{\boxed{P^{old}(j \mid x^n)}} \frac{P^{new}(j) p^{new}(x^n \mid j)}{p^{old}(x^n) P^{old}(j \mid x^n)} \right] \right\}}^{\boxed{\ln\left\{\sum_{j=1}^{M} l_j y_j\right\}}}$$

by Jensen's inequality:

$$\le -\sum_{n=1}^{N} \underbrace{\sum_{j=1}^{M} P^{old}(j \mid x^n) \ln \left\{ \frac{P^{new}(j) p^{new}(x^n \mid j)}{p^{old}(x^n) P^{old}(j \mid x^n)} \right\}}_{\boxed{\sum_{j=1}^{M} l_j \ln\{y_j\}}}$$

Back to EM

Change in the error:

$$E^{new} - E^{old} =$$

$$= -\sum_{n=1}^{N} \overbrace{\ln\left\{\sum_{j=1}^{M}\left[\underbrace{\boxed{P^{old}(j \mid x^n)}}_{\lambda}\frac{P^{new}(j)p^{new}(x^n \mid j)}{p^{old}(x^n)P^{old}(j \mid x^n)}\right]\right\}}^{\boxed{\ln\left\{\sum_{j=1}^{M} l_j y_j\right\}}}$$

by Jensen's inequality:

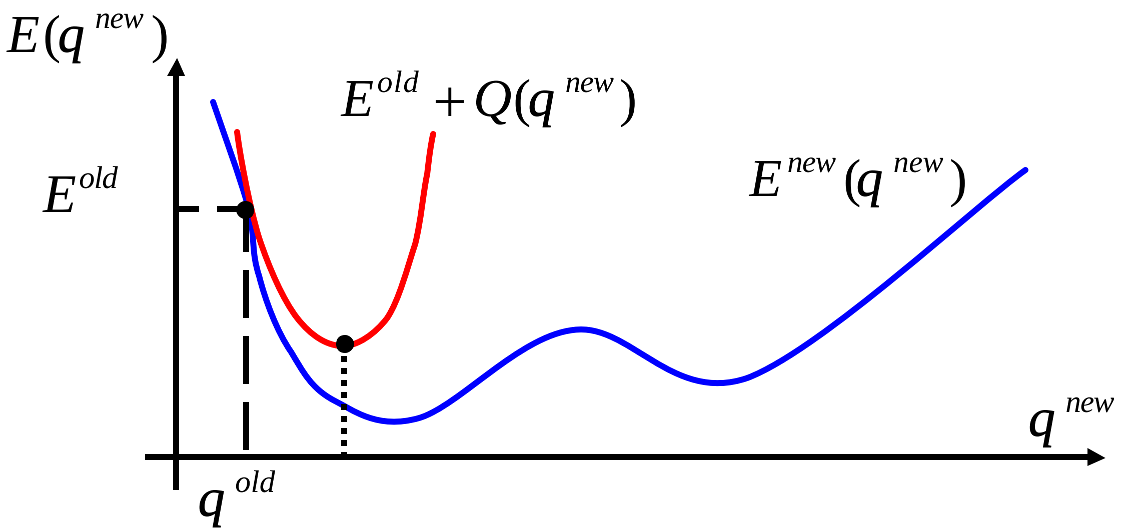
$$\leq \underbrace{\boxed{-\sum_{n=1}^{N}\sum_{j=1}^{M} P^{old}(j \mid x^n)\ln\left\{\frac{P^{new}(j)p^{new}(x^n \mid j)}{p^{old}(x^n)P^{old}(j \mid x^n)}\right\}}}_{\text{call this "}Q\text{"}}$$

EM as Upper Bound Minimization

Then: $E^{new} \leq E^{old} + Q$ - *upper bound* on $E^{new}(\theta^{new})$

Some observations:

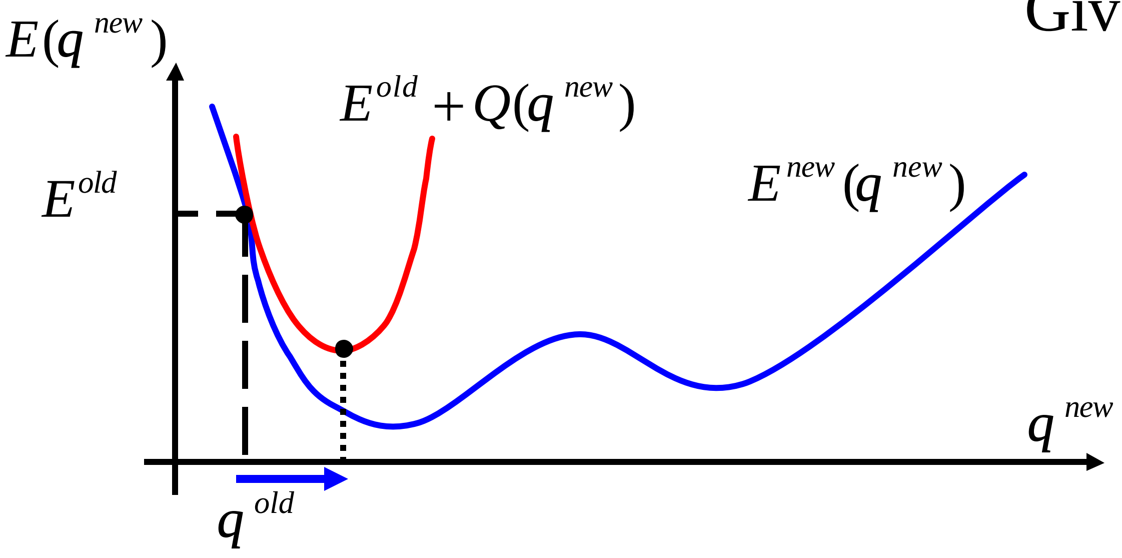
- Q is convex
- Q is a function of new parameters θ^{new}
- So is E^{new}
- If $\theta^{new} = \theta^{old}$ then $E^{new} = E^{old} + Q$



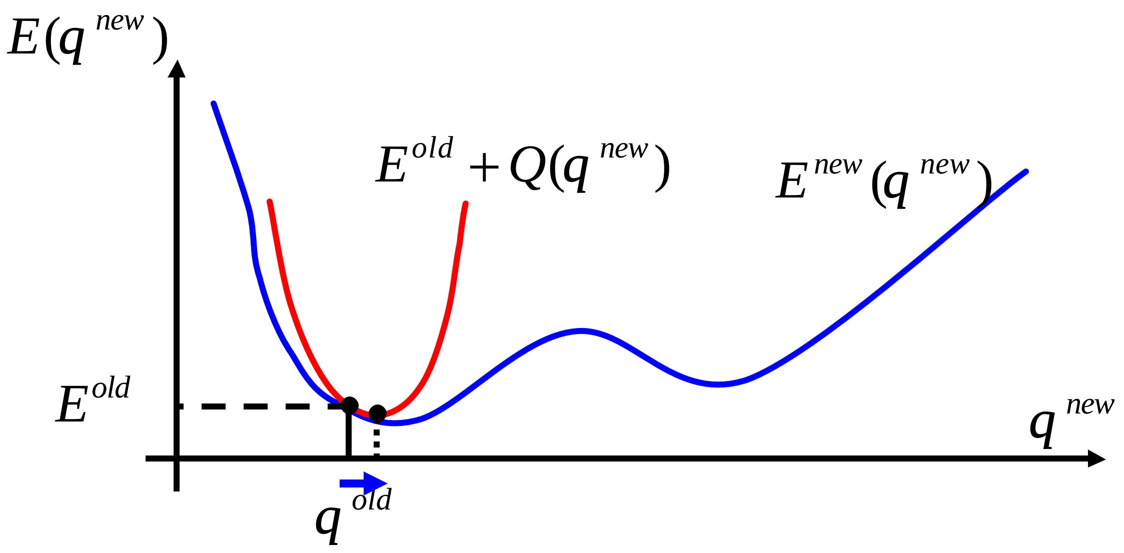
Step downhill in Q leads downhill in E^{new} !!!

EM Iteration

Given initial θ minimize Q



Compute new $E^{old} + Q$



EM (cont.)

$$Q = -\sum_{n=1}^{N}\sum_{j=1}^{M} P^{old}(j \mid x^n) \ln \left\{ \frac{P^{new}(j)\, p^{new}(x^n \mid j)}{\boxed{p^{old}(x^n) P^{old}(j \mid x^n)}} \right\}$$

Can drop these

$$\tilde{Q} = -\sum_{n=1}^{N}\sum_{j=1}^{M} P^{old}(j \mid x^n) \ln \left\{ P^{new}(j)\, p^{new}(x^n \mid j) \right\}$$

for a Gaussian mixture:

$$= -\sum_{n=1}^{N}\sum_{j=1}^{M} P^{old}(j \mid x^n) \left\{ \ln P^{new}(j) - \ln\left(G_j(x^n) \right) \right\}$$

As before – differentiate, set to 0, solve for parameter.

EM (cont.)

Straight-forward for means and covariances:

$$\hat{m}_j = \frac{\sum_{n=1}^{N} P^{old}(j \mid x^n) x^n}{\sum_{n=1}^{N} P^{old}(j \mid x^n)}$$

- convex sum, weighted w.r.t. previous estimate

$$\hat{\mathbf{S}}_j = \frac{\sum_{n=1}^{N} P^{old}(j \mid x^n)\left(x^n - \hat{m}_j\right)\left(x^n - \hat{m}_j\right)^T}{\sum_{n=1}^{N} P^{old}(j \mid x^n)}$$

- convex sum, weighted w.r.t. previous estimate

EM (cont.)

Need to enforce sum-to-one constraint for *P*(*j*):

$$J_P = \tilde{Q} + l\left(\sum_{j=1}^{M} P^{new}(j) - 1\right)$$

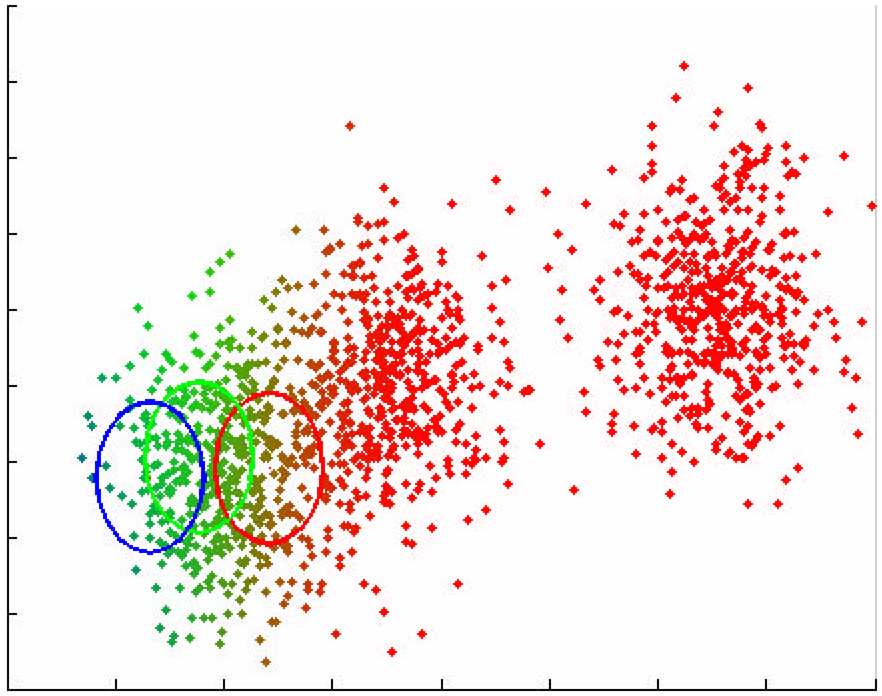
$$\frac{\partial}{\partial P^{new}(j)} J_P = -\sum_{n=1}^{N} \frac{P^{old}(j \mid x^n)}{P^{new}(j)} + l = 0$$

$$\Rightarrow \; l\, P^{new}(j) = \sum_{n=1}^{N} P^{old}(j \mid x^n)$$

$$\Rightarrow \; l \sum_{j=1}^{M} P^{new}(j) = \sum_{n=1}^{N} \sum_{j=1}^{M} P^{old}(j \mid x^n)$$

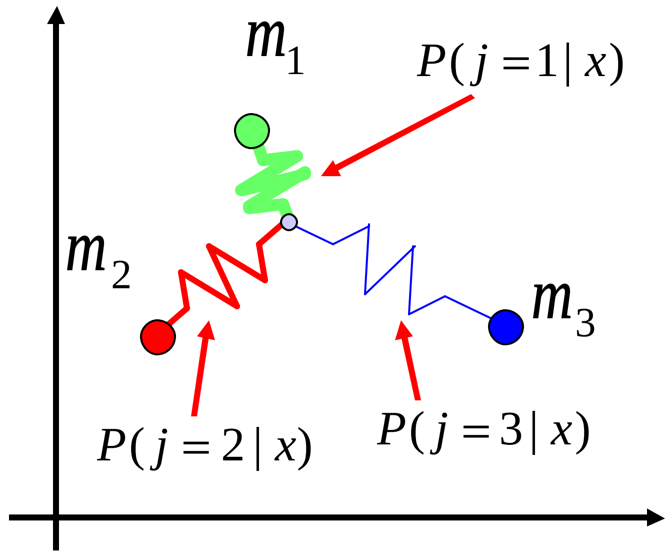
$$\Rightarrow \; l = N \qquad\qquad \Rightarrow \; P^{new}(j) = \frac{1}{N} \sum_{n=1}^{N} P^{old}(j \mid x^n)$$

EM Example



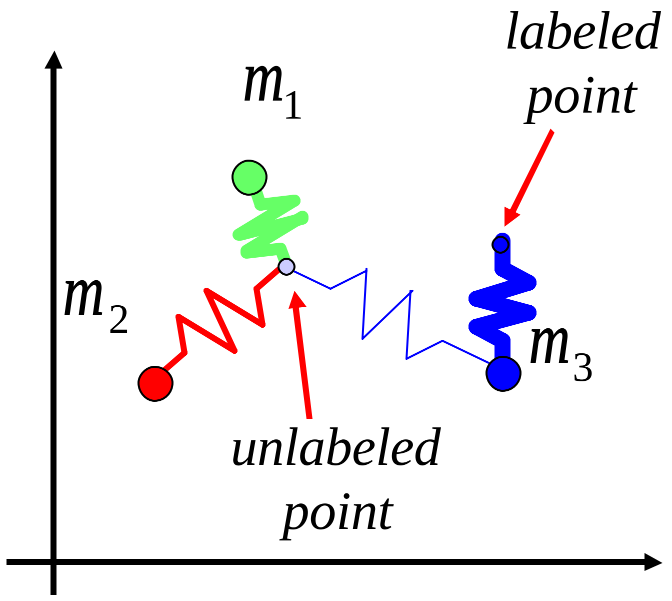
$N_c = 3$

EM Illustration



P(*j*|*x*) tells how much the data point affects each cluster, unlike in K-means.

You can manipulate *P*(*j*|*x*).
Eg: Partially labeled data



EM vs K-Means

Furthermore, $P(j|x)$ can be replaced with:

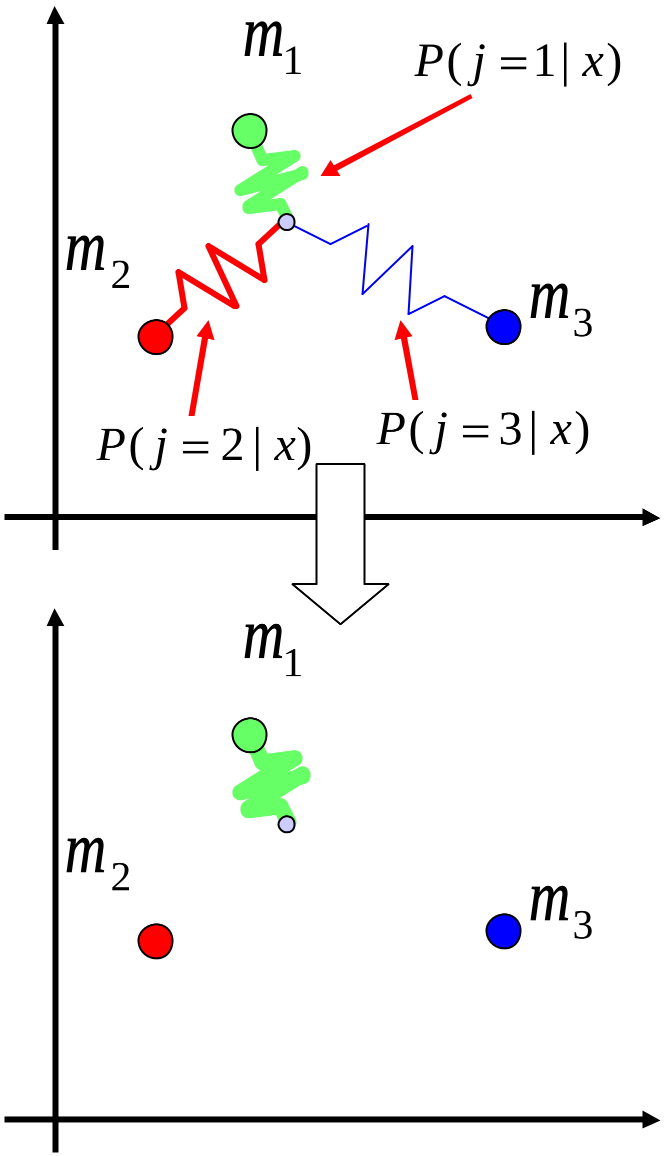
$$\tilde{P}(j|x) = \left. \frac{P(j|x)e^{g\,P(j|x)}}{\sum_k P(k|x)e^{g\,P(k|x)}} \right|_{g=0}$$

$$\text{if } \; g=0, \quad \tilde{P}(j|x) = P(j|x)$$

Now let's relax g :

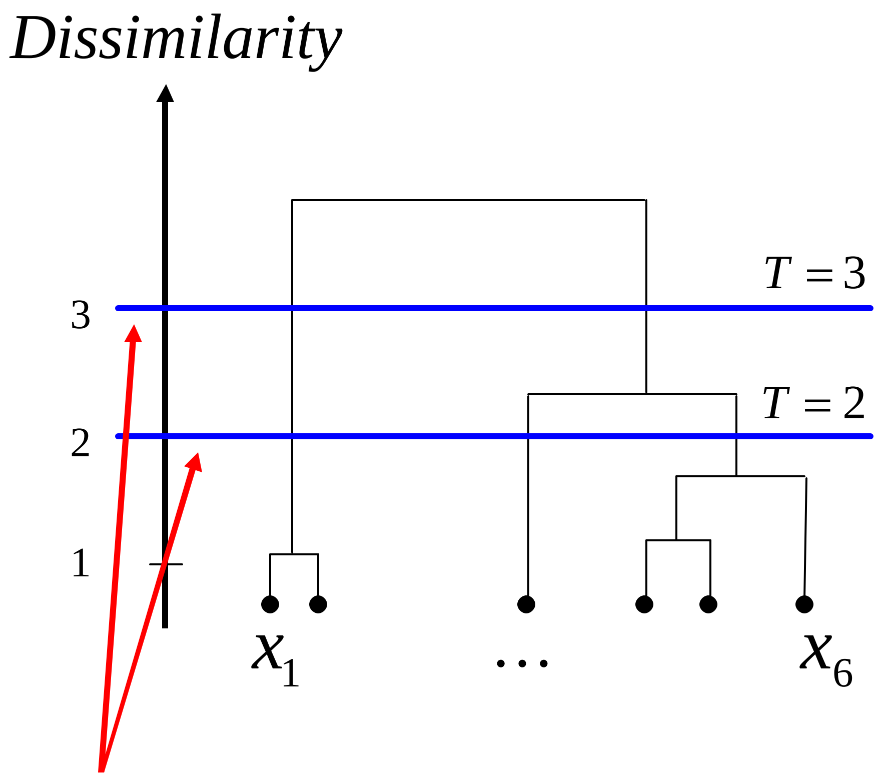
$$\lim_{g\to\infty} \tilde{P}(j|x) = \delta\left(P(j|x), \max P(j|x)\right)$$

This is K-Means!!!



Hierarchical Clustering

Ex: Dendrogram



There are 2 ways to do it:

- Agglomerative (bottom-up)
- Divisive (top-down)

Different thresholds induce different cluster configurations.

Stopping criterion – either a number of clusters, or a distance threshold

Hierarchical Agglomerative Clustering

General structure:

```
Initialize: $K,\ \hat{K} \leftarrow N,\ D_n \leftarrow x_n, n = 1..N$
        do      $\hat{K} \leftarrow \hat{K} - 1$
                $i, j = \operatorname{argmin}_{l,m} \boxed{d(D_l, D_m)}$
                $merge(D_i, D_j)$
        until   $\hat{K} == K$
```

Need to specify

Ex:

$$d = d_{mean}(D_i, D_j) = \left\| m_i - m_j \right\|$$

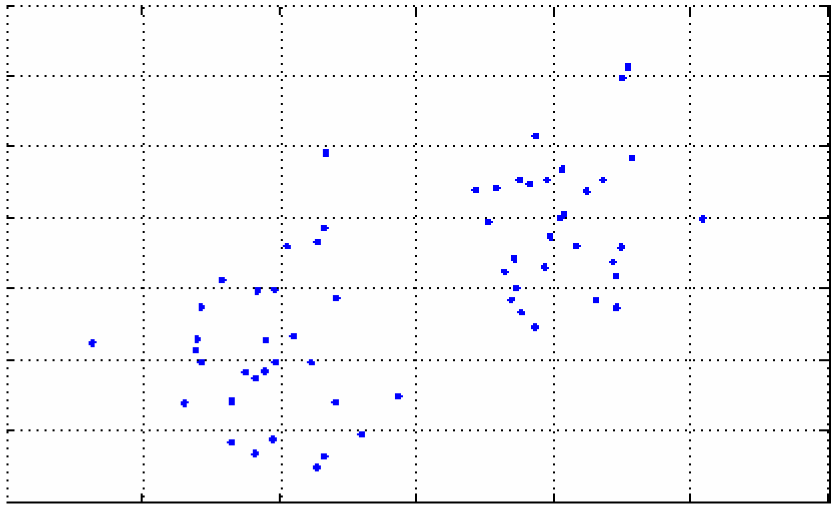
$$d = d_{\min}(D_i, D_j) = \min_{x_1 \in D_i, x_2 \in D_j} \left\| x_1 - x_2 \right\|$$

$$d = d_{\max}(D_i, D_j) = \max_{x_1 \in D_i, x_2 \in D_j} \left\| x_1 - x_2 \right\|$$

Each induces different algorithm

Single Linkage Algorithm

Choosing $d = d_{min}$ results in a Nearest Neighbor Algorithm (a.k.a single linkage algorithm, a.k.a. minimum algorithm)



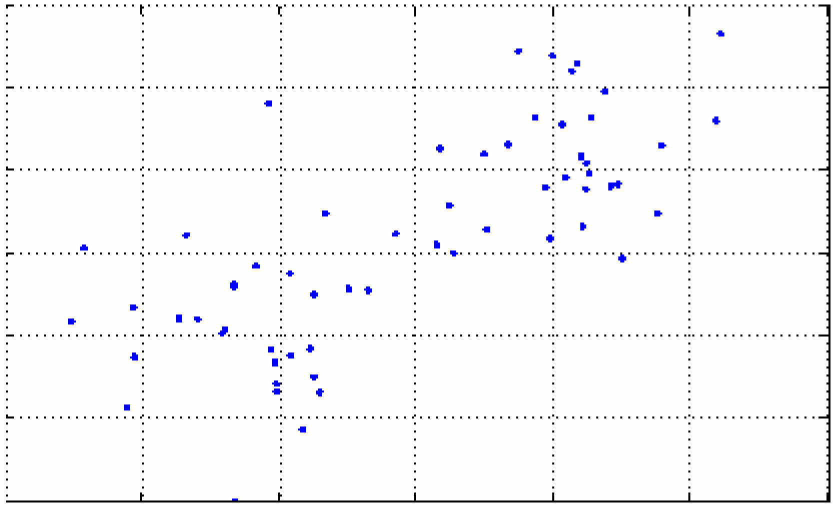
$N = 2$

Each cluster is a minimal spanning tree of the data in the cluster.

Identifies clusters that are well separated

Complete Linkage Algorithm

Choosing $d = d_{max}$ results in a Farthest Neighbor Algorithm (a.k.a. complete linkage algorithm, a.k.a. maximum algorithm)



$N = 2$

Each cluster is a complete subgraph of the data.

Identifies clusters that are well localized

Summary

- General concerns about choice of similarity metric
- K-means algorithm – simple but relies on Euclidean distances
- IsoData – old-school step towards model selection
- EM – “statistician’s K-means” – simple, general and convenient
- Some hierarchical clustering schemes