clustering: sets of instances 'close' together

- no class to be predicted - try to find "natural" clusters
  - clusters could be **exclusive, overlapping, probabilistic,** or **hierarchical**
  - nature and number of clusters depends on (unknown) domain mechanisms

d: $X \times X \rightarrow R$, such that for any x, y in X
$d(x,y) \geq 0$, $d(x,y) = d(y,x)$, $\underline{d(x,y) = 0 \Leftrightarrow x = y}$,
$d(x,y) \leq d(x,z) + d(z,y)$ for any z.

eg Euclidean distance between 2 places:

$$d(x,\ y) = \sqrt[2]{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}$$

$\underline{d(x,y) = 0 \Leftrightarrow x = y}$: if this is not satisfied, it's a **pseudo metric**. Ok for clustering!!
examples of distances: **Hamming distance:** metric between n-tuples of 0's and 1's,
= # of positions where 2 n-tuples have different entries.
• distance used in conceptual clusterers is similar to Hamming distance.

clustering, cont.

pseudo metric for conceptual clustering:

d(x,y) = # of functions f in description language for which $f(x) \neq f(y)$

start with empty set of clusters
while there are training data
    x := next instance
    if there is a cluster C with points near x
    then
        add x to C
        if C is unbalanced (considering all other clusters)
        then subdivide C into new clusters C_1 and C_2
    else
        form a new cluster whose only point is x

this leads to **extensional** descriptions of clusters,
ie just a particular family of subsets of training set.

clustering, cont.

- raw clusters are extensional descriptions
  - clusters are identified only by stating which example is in which cluster
- intensional description
  - assume there is some class $\{x \mid p(x)\}$ where p is a predicate true of all x in cluster C and false for all others. (p must be at least a sufficient condition!)
  - **Such a predicate p is crucial to assign new instances to clusters without doing the clustering all over again**!!!
  - Intensional descriptions are usually invented through **generalization** (clusters are just raw material for generalization)
  - given extensionally described C, c := point in C for which max $\{d(x,c) \mid x \in C\}$ is least (c is a **prototype** for C). D := max $\{d(x,c) \mid x \in C\}$. A **simple Intensional description** for corresponding class: $\{x \mid d(x,c) \leq D\}$

the simple intensional description given here can be easily computed if there are prototypes, and it does its job, ie it can assign new instances to clusters without repeating the entire clustering process.... usually, finding an intensional description is much harder and requires generalization methods....

overall quality of partition of instances into clusters

category utility CU measures how useful the clusters are in predicting values of attributes of instances; estimate of value *v* of attribute *a* of some instance should be better if we know that the instance is in some cluster *C* than if we don't know that (or else the clusters are useless!)

$$CU(C_1, ..., C_k) = \frac{\sum_{l=1}^{k} \Pr[C_l] \sum_i \sum_j (\Pr[a_i = v_{ij} | C_l]^2 - \Pr[a_i = v_{ij}]^2)}{k}$$

category utility is used to decide whether a category needs to be split up or fused with another...

dividing by k prevents 'overfitting', i.e. getting max CU by putting each instance into its own cluster!

basic clustering with k-means

randomly choose **k** points as cluster centers;

- assign instances to their nearest cluster center, using Euclidean distance;

compute means, i.e. centroid of instances in each cluster;

centroids are new cluster centers;

repeat • until cluster centers stabilize

- local optimum only
- different results may be due to initial random center choice
- poor choice of k may leave some clusters empty
- k-means has **many** variations ...

EM: Expectation Maximization

used for

- maximum likelihood estimation

- learning from incomplete examples

- learning Hidden Markov models

- dealing with Chicken and Egg problems…

EM: Expectation Maximization

we want solution for $$x^5 - 3x^2 + 2x - 17 = 0$$

rewrite as $$x = (3x^2 - 2x + 17)^{\frac{1}{5}}$$ and **assume** x is 1 ….

1.7826	←	1
1.8716	←	1.7826
1.8845	←	1.8716
1.8864	←	1.8845
1.8866	←	1.8864
1.8867	←	1.8866

not much change anymore ….

EM, cont.

- waiting for a bus, schedule unknown
- every second, 1/600 chance of bus coming

  waiting time is given by exponential density with $\mu = 10$ minutes: $\frac{1}{\mu} e^{-t/\mu}$

but we don't know μ, so we need some data ...

day 1: 7 min
day 2: 12 min
day 3:15 min
day 4: 10 min

max likelihood μ = (7+12+15+10) / 4 = 11 min

EM, cont.

suppose we were impatient:

day 1: waited 7 min and caught bus
day 2: waited 12 min and gave up
day 3: waited 8 min and caught bus
day 4: waited 5 min and gave up

1) start with **some** guess of μ, say 8 min.
2) fill in missing data with expected value **as if guess were correct**, i.e. add μ to values for missing data (days 2 and 4).
new data: 7, 20, 8, 13.
3) compute ML $\mu = (7+20+8+13) / 4 = 12$.
go to 2)

EM, cont.

in general:

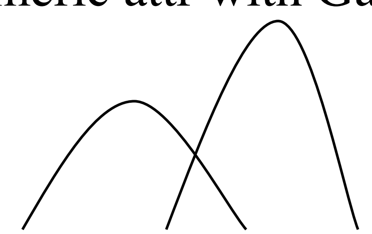
start by guessing missing info or guess some hypothesis, then repeat:

E. compute expected value of missing info given h (compute distribution of missing info)

M. compute max likelihood h given the guess (compute h that maximizes expected (log) likelihood over this distribution)

probabilistic clustering

- overall goal: find most likely set of clusters, given data and apriori assumptions
- instead of putting an instance firmly into a cluster, assign it a prob of belonging to <u>each</u> cluster
- finite mixture model:
  - k prob distributions (1 per cluster); each gives prob that an instance has certain attr values if it were known to be in that cluster, and 1 prob distribution for the clusters
  - simplest case: 1 numeric attr with Gaussian distribution for each cluster (but different μ and σ).



Clustering problem: Given {instances} and prespecified # of clusters, find each cluster's μ and σ, and population distribution.

probabilistic clustering, cont.

- if we knew from which distribution each instance comes, we could easily estimate μ (Σx_i / n) and σ^2 ($\Sigma(x_i - \mu)^2$ / (n-1)) for each distribution and Pr[distribution] itself.

but we don't know this!

- if we knew μ_A, σ_A, Pr(A), then the prob that instance x comes from distribution A (belongs to cluster A) is Pr[A | x] = (Pr[x | A] * Pr[A]) / Pr[x], i.e.

$$\frac{\frac{1}{(\sqrt{2\pi})\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}}{\Pr[x]}$$

normal distribution function for A

but we don't know this!

- our problem: some of the instance variables are **hidden**, **unobservable**. We don't know from <u>which</u> distribution x_i comes - so each instance is a k+1-tuple <x_i, z_{i1},..z_{ik}>, where z_{ij} =1 if x_i comes from the jth Normal distribution, else 0.

expectation maximization: EM

- EM looks for **maximum likelihood hypothesis**, i.e. *h* that maximizes Pr[data | *h*]
  - in k-means problem (assume all σ the same and known), EM looks for ML $h = [\mu_1,\ldots,\mu_k]$.

initialize $h = [\mu_1,\ldots,\mu_k]$ arbitrarily;

$\mu_{ML}=\text{argmin}_\mu \Sigma_{data}(x_i - \mu)^2$

repeat

assuming $h = [\mu_1,\ldots,\mu_k]$ is correct, compute $E[z_{ij}]$ for each z_{ij};

assuming $z_{ij} = E[z_{ij}]$, replace *h* by new ML $h' = [\mu_1',\ldots,\mu_k']$;

until convergence to a stationary value for *h*

$$E[z_{ij}] = \frac{\Pr[x = x_i | \mu = \mu_j]}{\sum_{n=1}^{k} \Pr[x = x_i | \mu = \mu_k]} = \frac{e^{-\frac{1}{2\sigma^2}(x_i-\mu_j)^2}}{\sum_{n=1}^{k} e^{-\frac{1}{2\sigma^2}(x_i-\mu_j)^2}}$$

new μ_j = $\Sigma_{data} E[z_{ij}]x_i / \Sigma_{data} E[z_{ij}]$

EM, cont.

- k-means stops when the classes of instances don't change from one iteration to the next but
- EM may converge toward fixed point, never gets there
  - how close are we? Calculate **likelihood** that data come from this data set:
    $\prod_i \left(\Pr(\text{Cluster}_1)\Pr(x_i \mid \text{Cluster}_1) + \Pr(\text{Cluster}_2)\Pr(x_i \mid \text{Cluster}_2) + \ldots\right)$
  - this measures how good the clustering is, and increases with each iteration
  - for speed, sum logs of all components instead: **log-likelihood**
- iterate until increase in log-likelihood becomes negligible
  - e.g. until difference between successive values of log-likelihood $< 10^{-10}$ for 10 iterations
  - **EM starts with 1 cluster, and adds new clusters until estimated log-likelihood decreases. (EM tries to maximize log-likelihood of future data via cross validation.)**

EM, cont.

- EM works in this situation
  - $X = \{x_1,\ldots,x_m\}$ (observed data in m instances); $Z = \{z_1,\ldots, z_m\}$ (unobserved data in these instances); $Y = X \cup Z$ (the full data); (Z, Y are random variables, with prob distribution depending on θ and X)
  - estimate parameters θ that describe prob distribution governing Y
- find ML h' that maximizes **E[ln Pr(Y | *h*')]**

- **estimate:** using h instead of θ and X, estimate prob distribution over Y,

  $$E[\ln \Pr(Y \mid h') \mid h, X]$$

- **maximize:** replace h by h' that maximizes $E[\ln \Pr(Y \mid h') \mid h, X]$

may get trapped at local optimum