Machine learning: Part 2

- Unsupervised Learning: k-means algorithm
- Learning from incomplete Data: EM algorithm

*Slides based on those of D. Poole and A. Mackworth

Unsupervised learning

- Unsupervised learning is the problem of identifying multiple categories in a collection of objects.
- The problem is unsupervised because the category labels are not given.
- *e.g.*, classification of stars by astronomers
- In hard clustering, each example is placed definitively in a class.
- In soft clustering, each example has a probability distribution over its class.

EM Algorithm for clustering

- Start with a random theory or randomly classified data
- Repeat the following two steps:
  - E-step generates the expected classification for each example.
  - M-step generates the best theory using the current classification of data.
- We consider two instances of the EM algorithm

k-means algorithm

Used for hard clustering.

Inputs:

- training examples
- the number of classes, k

Outputs:

- a prediction of a value for each feature for each class
- an assignment of examples to classes

k-means algorithm formalized

- E is the set of all examples
- the input features are $X_1, \ldots, X_n$
- $val(e, X_j)$ is the value of feature X_j for example e.
- there is a class for each integer $i \in \{1, \ldots, k\}$.

The k-means algorithm outputs

- a function $class : E \to \{1, \ldots, k\}$.
  $class(e) = i$ means e is in class i.
- a $pval$ function where $pval(i, X_j)$ is the prediction for each example in class i for feature X_j.

k-means algorithm formalized

The sum-of-squares error for $class$ and $pval$ is

$$\sum_{e \in E} \sum_{j=1}^{n} (pval(class(e), X_j) - val(e, X_j))^2 .$$

Aim: find $class$ and $pval$ that minimize sum-of-squares error.

k-means algorithm

Initially, randomly assign the examples to the classes.
Repeat the following two steps:

- For each class i and feature X_j,

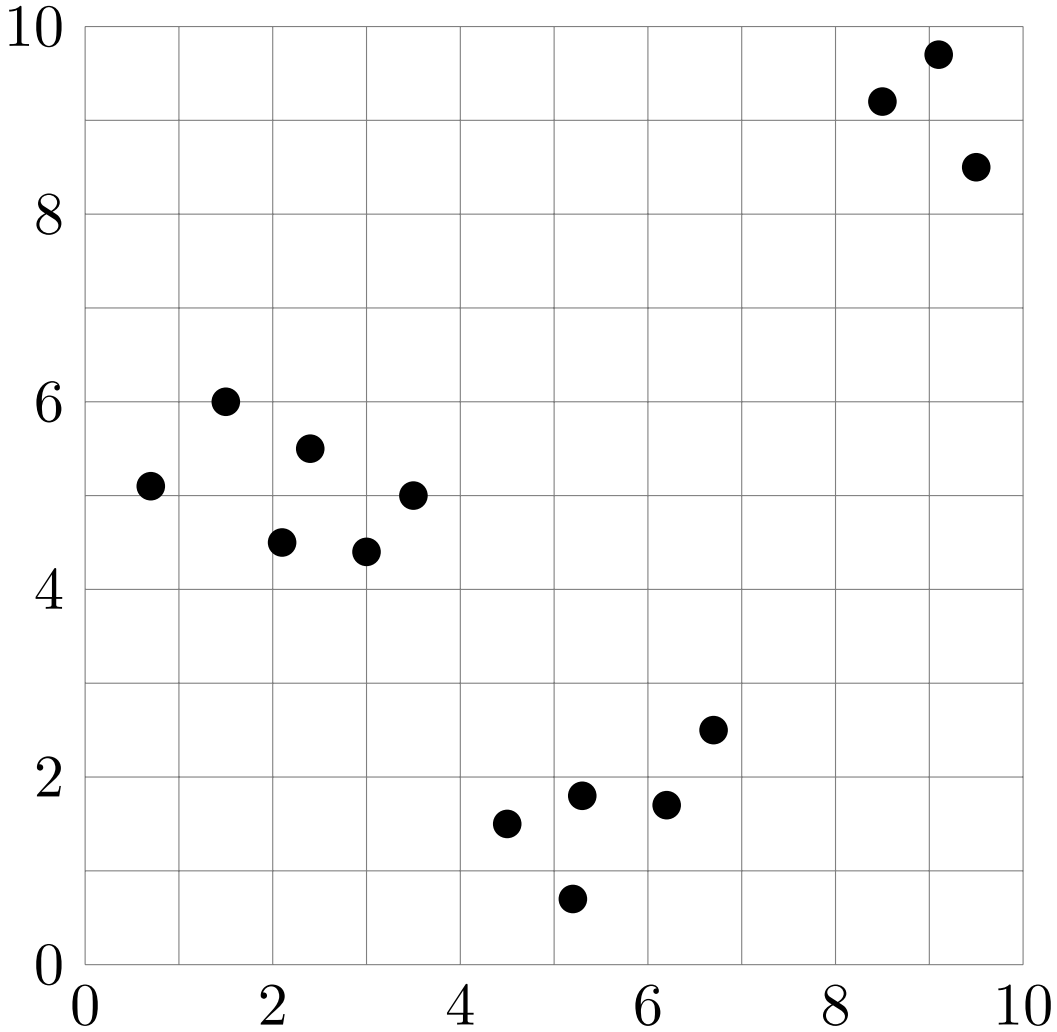
$$pval(i, X_j) \leftarrow \frac{\sum_{e:class(e)=i} val(e, X_j)}{|\{e : class(e) = i\}|},$$

- For each example e, assign e to the class i that minimizes

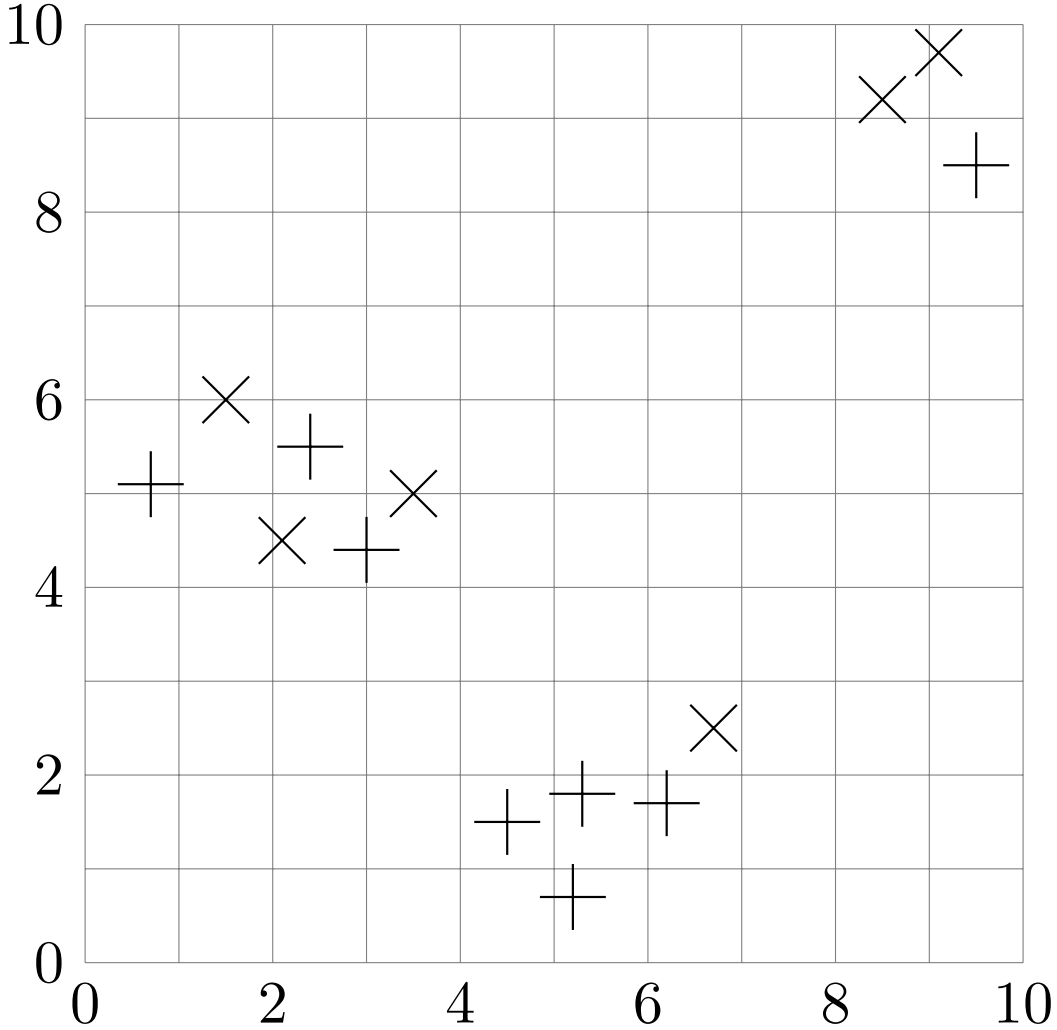
$$\sum_{j=1}^{n} (pval(i, X_j) - val(e, X_j))^2 .$$

until the second step does not change the assignment of any example.

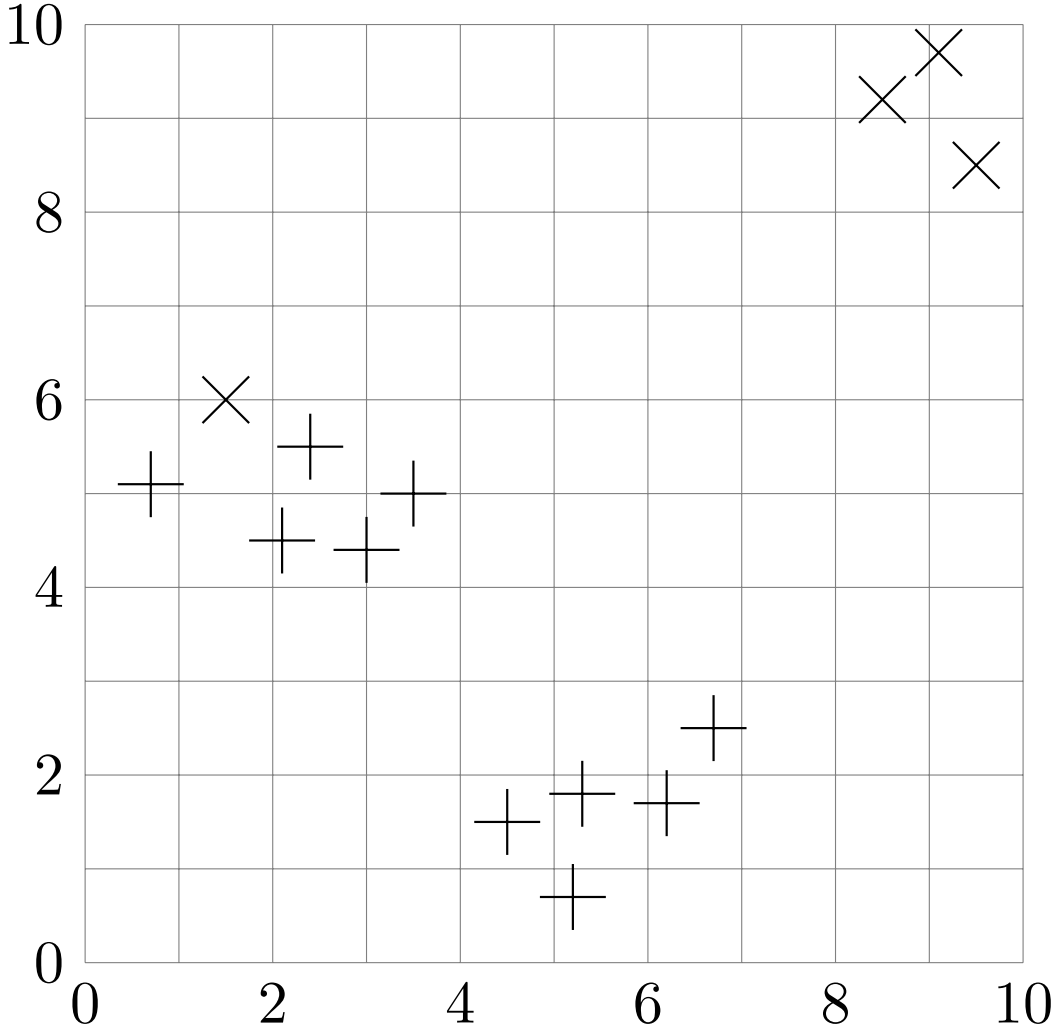
Example Data



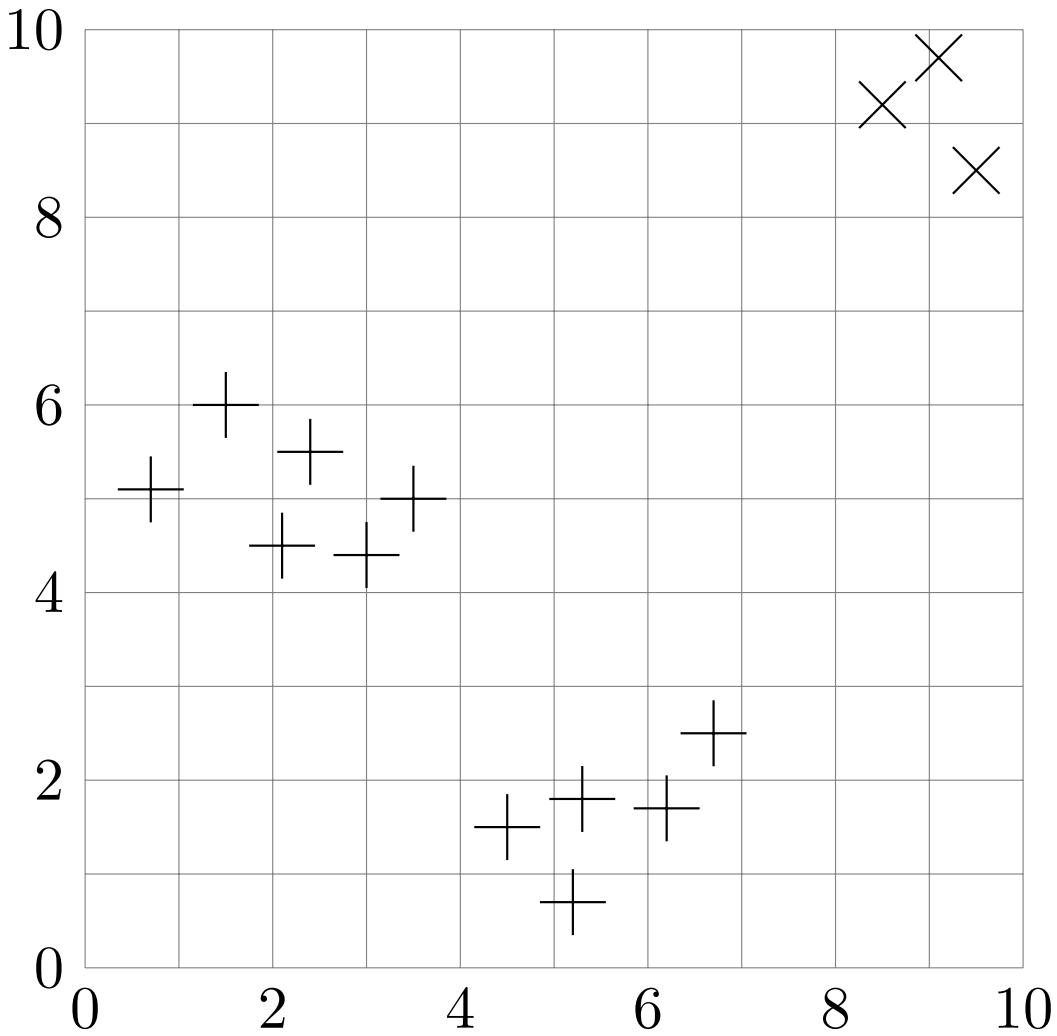
Random Assignment to Classes



Assign Each Example to Closest Mean



Ressign Each Example to Closest Mean



Stable assignment found

Other clustering results

- A different initial assignment can give different clustering.
- One clustering is for the lower points to be in one class, and for the other points to be in another class.
- Running the algorithm with three classes would separate the data into the top-right, the left-center, and the lower clusters.

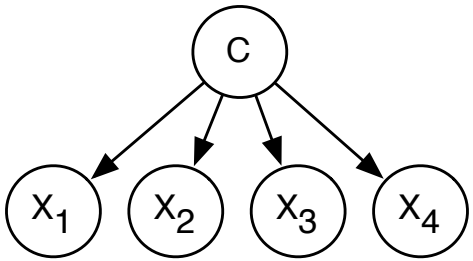
Properties of k-means

- This algorithm will eventually converge to a local minimum.
- It is not guaranteed to converge to a global minimum.
- Increasing k can always decrease error until k is the number of different examples.

EM algorithm

- Used for soft clustering — examples are probabilistically in classes.
- k-valued random variable C

Model



Data

X_1	X_2	X_3	X_4
t	f	t	t
f	t	t	f
f	f	t	t
		$\ldots$	

➪

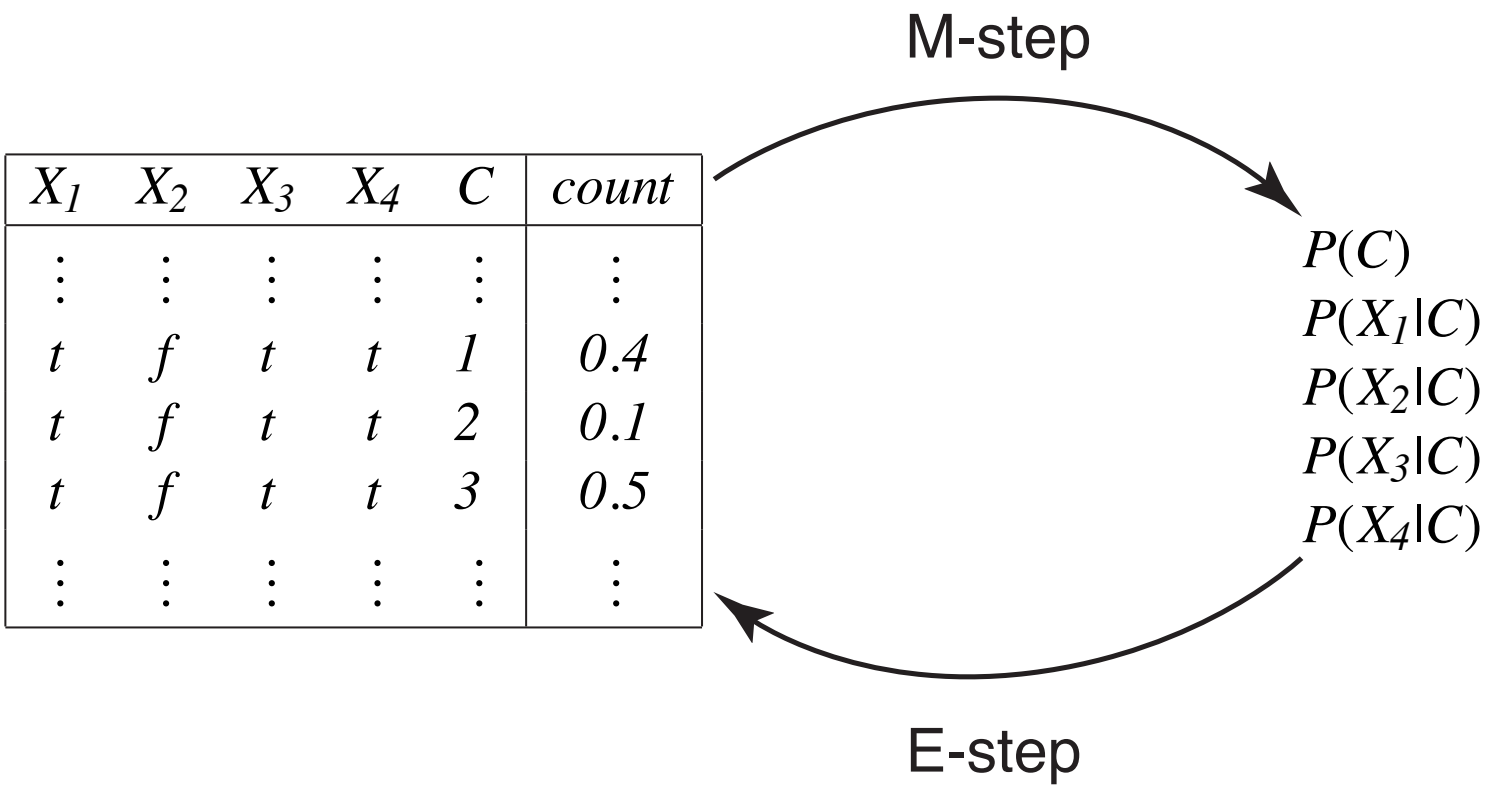
Probabilities

$P(C)$
$P(X_1|C)$
$P(X_2|C)$
$P(X_3|C)$
$P(X_4|C)$

EM Algorithm Overview

- Repeat the following two steps:
  - E-step give the expected number of data points for the unobserved variables based on the given probability distribution.
  - M-step infer the (maximum likelihood or maximum aposteriori probability) probabilities from the data.
- Start either with made-up data or made-up probabilities.
- EM will converge to a local maxima.

EM algorithm



Augmented data – E step

Suppose $k = 3$, and $dom(C) = \{1, 2, 3\}$.
$P(C = 1|X_1 = t, X_2 = f, X_3 = t, X_4 = t) = 0.407$
$P(C = 2|X_1 = t, X_2 = f, X_3 = t, X_4 = t) = 0.121$
$P(C = 3|X_1 = t, X_2 = f, X_3 = t, X_4 = t) = 0.472$:

X_1	X_2	X_3	X_4	*Count*
⋮	⋮	⋮	⋮	⋮
t	f	t	t	100
⋮	⋮	⋮	⋮	⋮

$\longrightarrow$

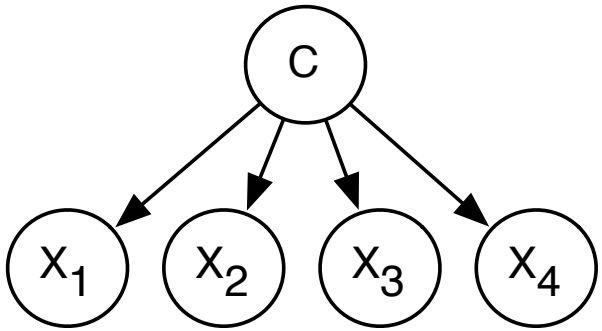
$A[X_1, \ldots, X_4, C]$

X_1	X_2	X_3	X_4	C	*Count*
⋮	⋮	⋮	⋮	⋮	⋮
t	f	t	t	1	40.7
t	f	t	t	2	12.1
t	f	t	t	3	47.2
⋮	⋮	⋮	⋮	⋮	⋮

M step

X_1	X_2	X_3	X_4	C	$Count$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
t	f	t	t	1	40.7
t	f	t	t	2	12.1
t	f	t	t	3	47.2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

$\longrightarrow$

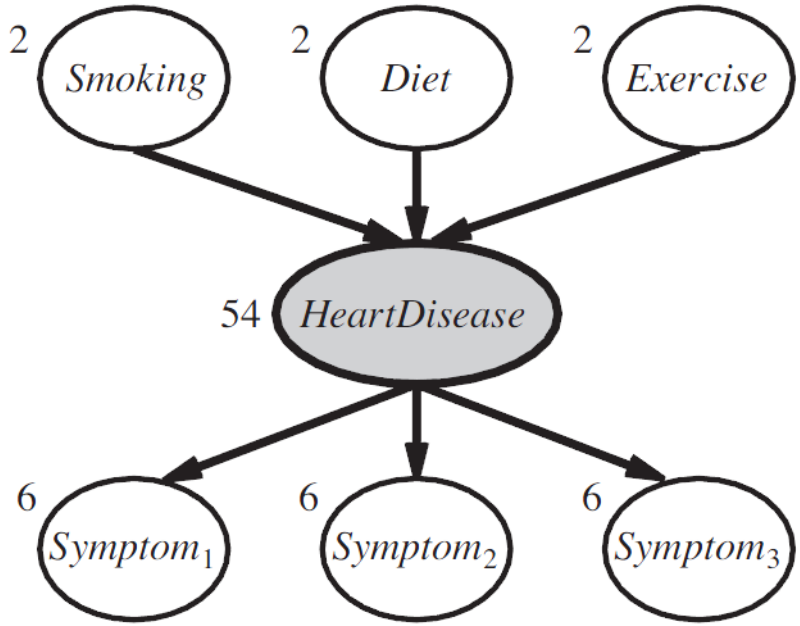


$$P(C{=}v_i) = \frac{\sum_{t \models C=v_i} Count(t)}{\sum_t Count(t)}$$

$$P(X_k = v_j | C{=}v_i) = \frac{\sum_{t \models C=v_i \wedge X_k=v_j} Count(t)}{\sum_{t \models C=v_i} Count(t)}$$

Learning with hidden variables

- Many real-world problems have hidden (a.k.a latent) variables



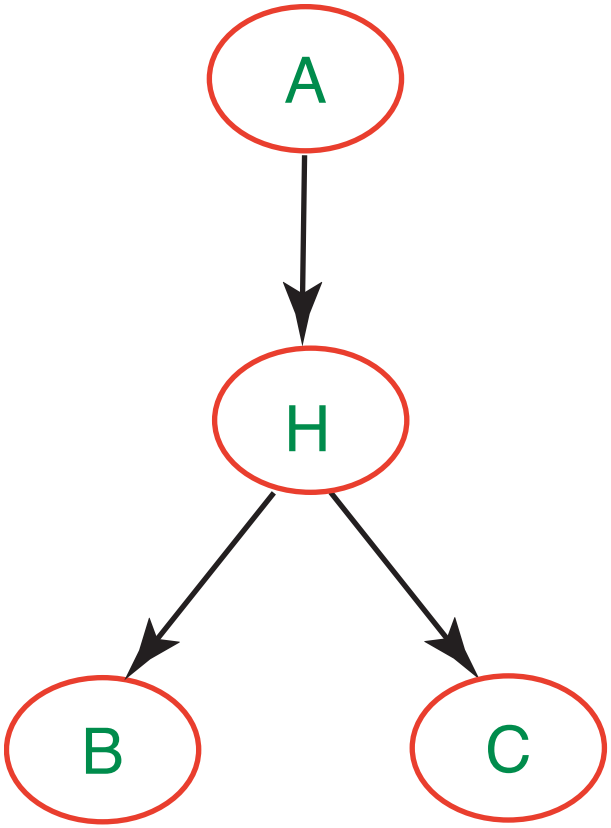
A simple diagnostic network for heart disease

- Hidden variables complicate the learning problem.

EM algorithm

- Repeat the following two steps:
  - E-step give the expected number of data points for the unobserved variables based on the given probability distribution. Requires probabilistic inference.
  - M-step infer the (maximum likelihood) probabilities from the data. This is the same as the fully observable case.
- Start either with made-up data or made-up probabilities.
- EM will converge to a local maxima.

A simple example



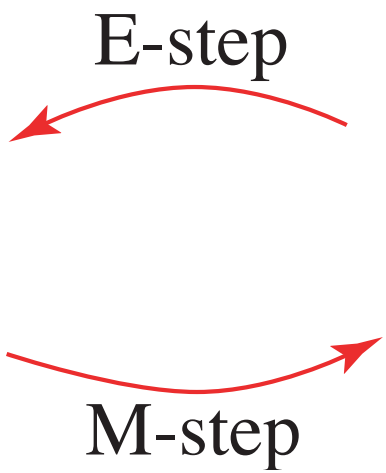
- What if we had only observed values for A, B, C?

A	B	C
t	f	t
f	t	t
t	t	f
	$\ldots$	

EM algorithm

Augmented Data

A	B	C	H	$Count$
t	f	t	t	0.7
t	f	t	f	0.3
f	t	t	f	0.9
f	t	t	t	0.1
	...			...



Probabilities

$P(A)$
$P(H|A)$
$P(B|H)$
$P(C|H)$