CS440/ECE448: Intro to Artificial Intelligence

Lecture 20
More on learning graphical models

Prof. Julia Hockenmaier
juliahmr@illinois.edu

http://cs.illinois.edu/fa11/cs440

Bayes Nets

A Bayes Net defines a joint distribution $P(X_1 \ldots X_n)$ over a set of random variables $X_1 \ldots X_n$

Using the chain rule, we can factor $P(X_1 \ldots X_n)$ into a product of *n* conditional distributions:

$$P(X_1 \ldots X_n) = \prod_j P(X_i \mid X_1 \ldots X_{i-1}).$$

A Bayes Net makes a number of (conditional) independence assumptions:

$$P(X_1 \ldots X_n) =_{def} \prod_j P(X_i \mid Parents(X_i) \subseteq \{X_{1 \ldots} X_{i-1}\})$$

Learning Bayes Nets

Parameter estimation: Given some data D over a set of random variables **X** and a Bayes Net (with empty CPTs) estimate the parameters (= fill in the CPTs) of the Bayes Net.

Structure learning: Given some data D over a set of random variables **X**, find a Bayes Net (define its CPTs) and estimate its parameters.
(This is much harder… we won't deal with it here)

Bayes Rule

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)}$$

$P(h)$: prior probability of hypothesis
$P(h \mid D)$: posterior probability of hypothesis.
$P(D \mid h)$: likelihood of data, given hypothesis

Prior ∝ posterior × likelihood

$$P(h \mid D) \propto P(D \mid h)P(h)$$

Three kinds of estimation techniques

Bayes optimal: Marginalize out the hypotheses

$$P(X \mid \boldsymbol{D}) = \textstyle\sum_i P(X \mid h_i) P(h_i \mid \boldsymbol{D})$$

MAP (maximum a posteriori):
Pick the hypothesis with the highest posterior

$$h_{MAP} = argmax_h\, P(h|D)$$

ML (maximum likelihood):
Pick the hypothesis that assigns highest likelihood

$$h_{ML} = argmax_h\, P(D|h)$$

Maximum likelihood learning

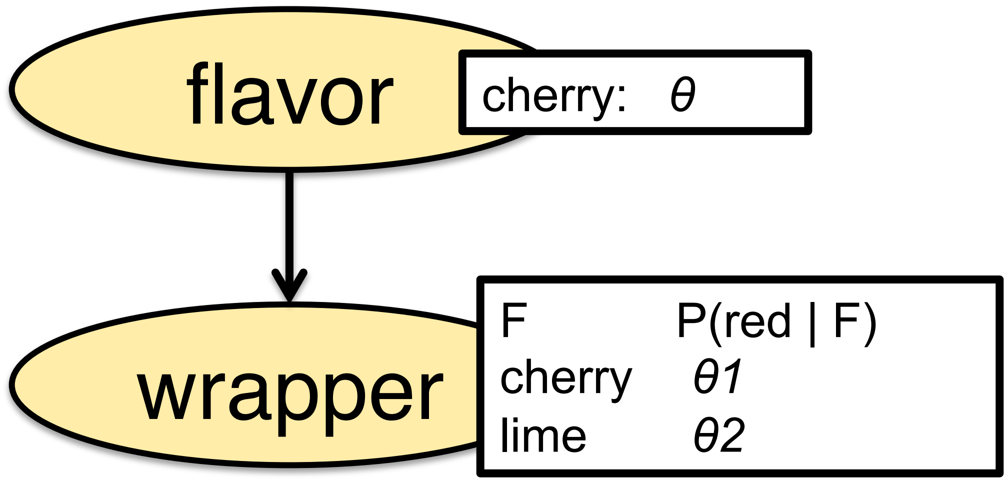
Given data ***D***, we want to find the parameters that maximize $P(\boldsymbol{D} \mid \theta)$.

We have a data set with N candies.
c are cherry. $l = (N\text{-}c)$, are lime.
Parameter θ = probability of cherry

Maximum likelihood estimate: $\theta = \mathrm{c/N}$

A more complex model

Now the candy has two kinds of wrappers (red or green).
The wrapper is chosen probabilistically, depending on the flavor of the candy.



Out of *N* candies, c are cherry. r_c are cherry with a red wrapper, r_l are lime with a red wrapper

The likelihood of this data set:

$$P(d \mid \theta, \theta_1, \theta_2) = \theta^{c}(1-\theta)^{N-c}\, \theta_1^{\,rc}\, (1-\theta_1)^{c-rc} \theta_2^{\,rl}\, (1-\theta_1)^{(N-c)-rl}$$

The log likelihood of this data set:

$$\begin{aligned} L(d \mid \theta, \theta_1, \theta_2) = & [c \log\theta + (N-c)\log(1-\theta)] \\ & +[r_c \log\theta_1 + (c-r_c)\log(1-\theta_1)] \\ & +[l_c \log\theta_2 + (N-c-l_c)\log(1-\theta_2)] \end{aligned}$$

The ML parameter estimates:

$$\theta = c/N \quad \theta_1 = r_c/c \quad \theta_2 = r_l/(N-c)$$

Medical diagnosis

Patients see a doctor and complain about a number of symptoms (headache, 100F fever, …).

What is the most likely disease d_i, given the set of symptoms S the patient has?

$$\arg\max_{d_i} P\left(d_i \mid \overline{S}\right)$$

The Naïve Bayes classifier

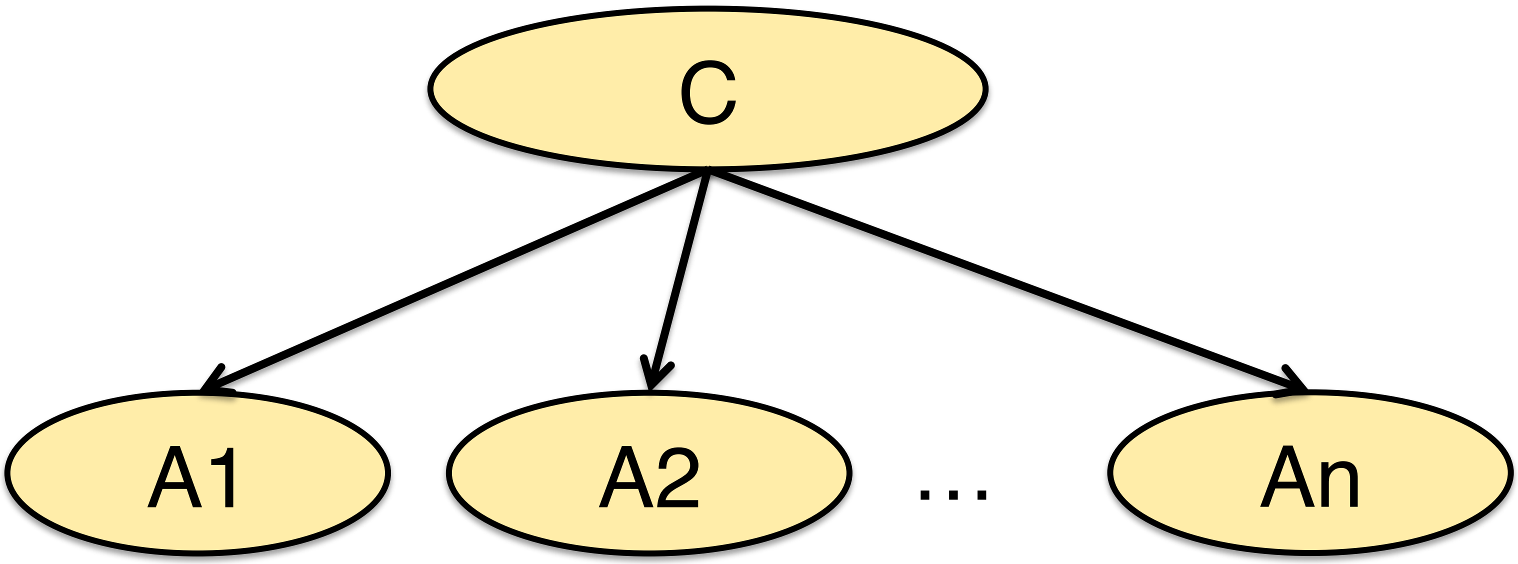
Assume the items in your data set have a number of attribute $A_1 \ldots A_n$.

Each item also belongs to one of a number of given classes $C_1 \ldots C_k$.

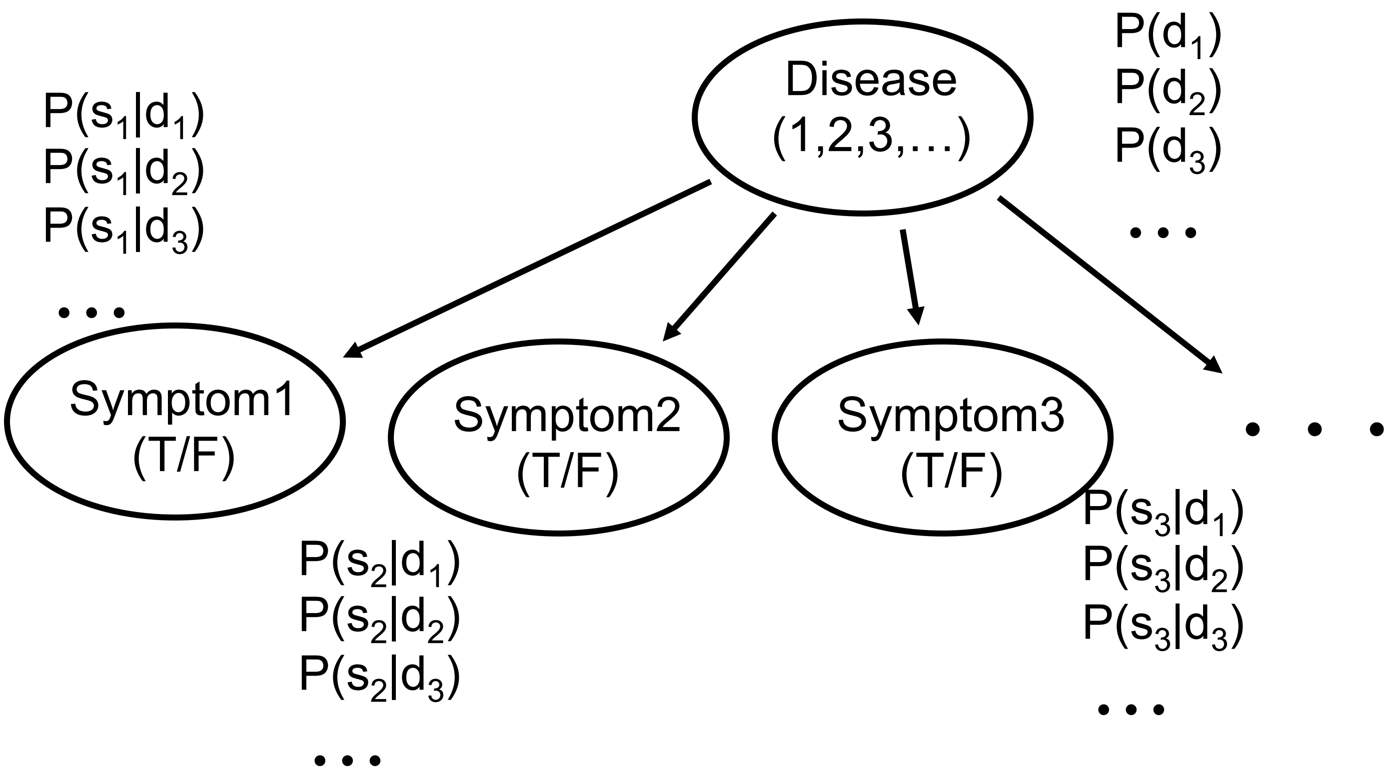
Which attributes an item has depends on its class.

If you only observe the attributes of an item, can you predict the class?

The Naïve Bayes classifier



Naïve Bayes


P(d1)
P(d2)
P(d3)
…
Disease
(1,2,3,…)
P(s1|d1)
P(s1|d2)
P(s1|d3)
…
Symptom1
(T/F)
Symptom2
(T/F)
Symptom3
(T/F)
. . .
P(s2|d1)
P(s2|d2)
P(s2|d3)
…
P(s3|d1)
P(s3|d2)
P(s3|d3)
…

Naïve Bayes

$$\text{argmax}_C\ P(C|\ A_1 \ldots A_n) =$$
$$= \text{argmax}_C\ P(A_1 \ldots A_n \mid C)\ P(C)$$
$$= \text{argmax}_C \prod_j P(A_j \mid C)\ P(C)$$

We need to estimate:

– the multinomial P(C)

– for each attribute A_j and class c $P(A_j \mid c)$

Maximum likelihood estimation

If we have a set of training data where the class of each item is given:

– the multinomial P(C=c) = freq(c)/N

– for each attribute A_j and class c:
  $P(A_j = a| c) = freq(a, c)/freq(c)$

where

freq(c) = the number of items in the training data that have class c

freq(a, c) = the number of items in the training data that have attribute a and class c.