

CS5010
Artificial Intelligence Principles:
Lecture 5

STATISTICAL LEARNING

CHAPTER 20, SECTIONS 1–3

Outline

- ◇ Bayesian learning
- ◇ Maximum *a posteriori* and maximum likelihood learning
- ◇ Bayes net learning
 - ML parameter learning with complete data
 - linear regression

Calculate the probability of each hypothesis given the data.
Use ML to make predictions on that basis.

Full Bayesian learning

View learning as Bayesian updating of a probability distribution over the **hypothesis space**

H is the hypothesis variable, values h_1, h_2, \dots , prior $\mathbf{P}(H)$

j th observation d_j gives the outcome of random variable D_j
training data $\mathbf{d} = d_1, \dots, d_N$

Given the data so far, each hypothesis has a posterior probability:

$$P(h_i|\mathbf{d}) = \alpha P(\mathbf{d}|h_i)P(h_i)$$

where $P(\mathbf{d}|h_i)$ is called the **likelihood**

Predictions use a likelihood-weighted average over the hypotheses:

$$\mathbf{P}(X|\mathbf{d}) = \sum_i \mathbf{P}(X|\mathbf{d}, h_i)P(h_i|\mathbf{d}) = \sum_i \mathbf{P}(X|h_i)P(h_i|\mathbf{d})$$

No need to pick one best-guess hypothesis!

Posterior
prob. of the
hypothesis
given the
evidence
=
prior prob.
of the
hypothesis
times the
likelihood
of the
evidence
given that
the
hypothesis
is true

The alpha term is a constant that accounts for the probability that the evidence is true

Example

X: next candy is lime

Suppose there are five kinds of bags of candies:

10% are h_1 : 100% cherry candies

20% are h_2 : 75% cherry candies + 25% lime candies

40% are h_3 : 50% cherry candies + 50% lime candies

20% are h_4 : 25% cherry candies + 75% lime candies

10% are h_5 : 100% lime candies

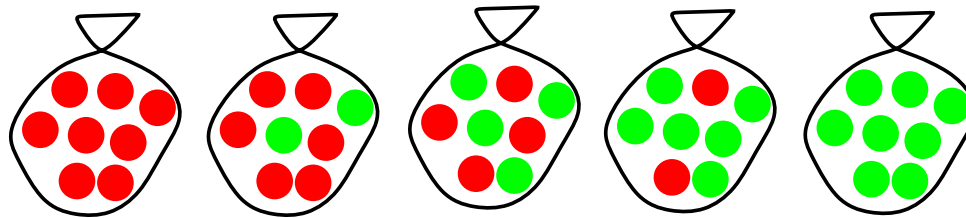
$$P(X|h_1) = 0$$

$$P(X|h_2) = 0.25$$

$$P(X|h_3) = 0.5$$

$$P(X|h_4) = 0.75$$

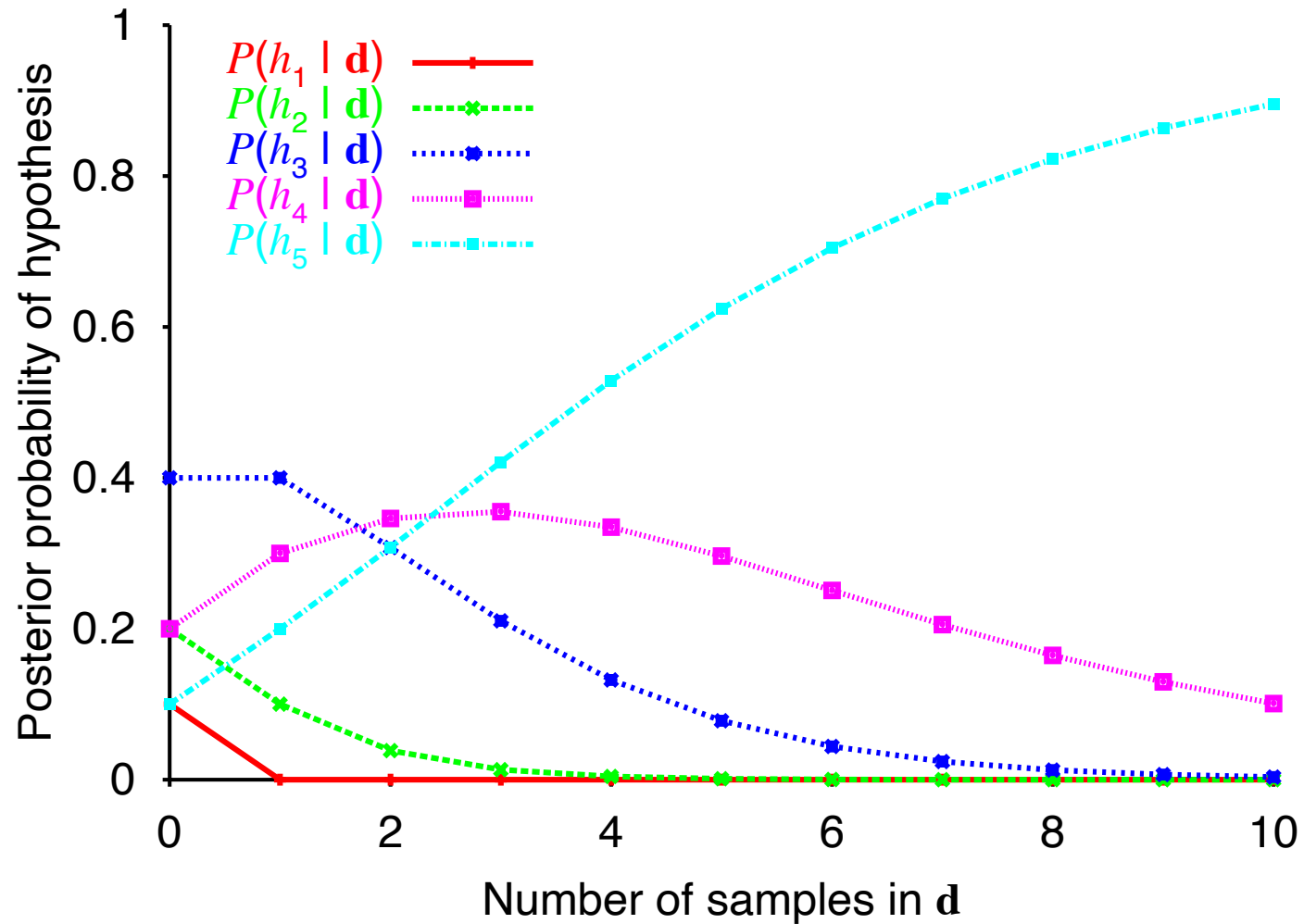
$$P(X|h_5) = 1$$



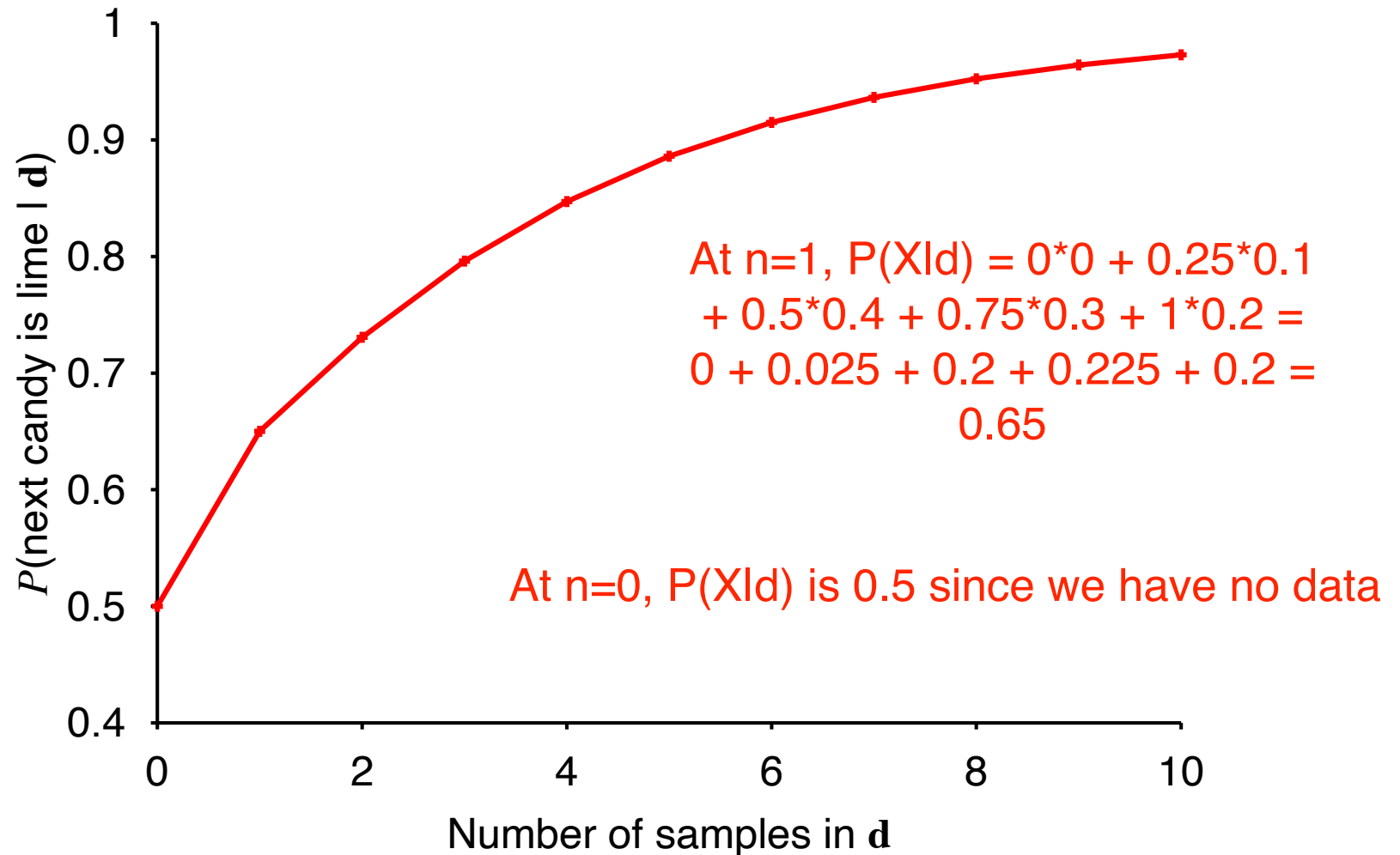
Then we observe candies drawn from some bag: ● ● ● ● ● ● ● ● ● ●

What kind of bag is it? What flavour will the next candy be?

Posterior probability of hypotheses



Prediction probability



Bayesian prediction eventually agrees with the correct hypothesis

MAP approximation

Summing over the hypothesis space is often intractable
(e.g., 18,446,744,073,709,551,616 Boolean functions of 6 attributes)

Maximum a posteriori (MAP) learning: choose h_{MAP} maximizing $P(h_i|\mathbf{d})$

I.e., maximize $P(\mathbf{d}|h_i)P(h_i)$ or $\log P(\mathbf{d}|h_i) + \log P(h_i)$

Log terms can be viewed as (negative of)

bits to encode data given hypothesis + bits to encode hypothesis

This is the basic idea of minimum description length (MDL) learning

For deterministic hypotheses, $P(\mathbf{d}|h_i)$ is 1 if consistent, 0 otherwise

\Rightarrow MAP = simplest consistent hypothesis (cf. science)

MAP is dangerous with sparse evidence. In the example, we'd select h_5 after three lime candies with probability 1. The actual Bayesian prediction has $P = 0.8$

As more data arrives the MAP & Bayesian probabilities converge

ML approximation

For large data sets, prior becomes irrelevant

Maximum likelihood (ML) learning: choose h_{ML} maximizing $P(\mathbf{d}|h_i)$

I.e., simply get the best fit to the data; identical to MAP for uniform prior (which is reasonable if all hypotheses are of the same complexity)

ML is the “standard” (non-Bayesian) statistical learning method

We assume a uniform prior over the hypothesis space.

No hypothesis is preferred before we start seeing data.

Enough data will swamp this prior distribution.

ML parameter learning in Bayes nets

Bag from a new manufacturer; fraction θ of cherry candies?

Any θ is possible: continuum of hypotheses h_θ

θ is a **parameter** for this simple (**binomial**) family of models

Suppose we unwrap N candies, c cherries and $\ell = N - c$ limes

These are **i.i.d.** (independent, identically distributed) observations, so

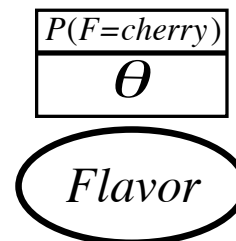
$$P(\mathbf{d}|h_\theta) = \prod_{j=1}^N P(d_j|h_\theta) = \theta^c \cdot (1 - \theta)^\ell$$

Maximize this w.r.t. θ —which is easier for the **log-likelihood**:

$$L(\mathbf{d}|h_\theta) = \log P(\mathbf{d}|h_\theta) = \sum_{j=1}^N \log P(d_j|h_\theta) = c \log \theta + \ell \log(1 - \theta)$$

$$\frac{dL(\mathbf{d}|h_\theta)}{d\theta} = \frac{c}{\theta} - \frac{\ell}{1 - \theta} = 0 \quad \Rightarrow \quad \theta = \frac{c}{c + \ell} = \frac{c}{N}$$

Seems sensible, but causes problems with 0 counts!



Maxima and minima occur at places where the derivative is zero

- Method: 1** Derive expression for likelihood of data as a function of parameters
- 2.** Obtain derivative of the log likelihood wrt each parameter
- 3.** Find parameter values where derivatives are zero

Multiple parameters

Red/green wrapper depends probabilistically on flavor:

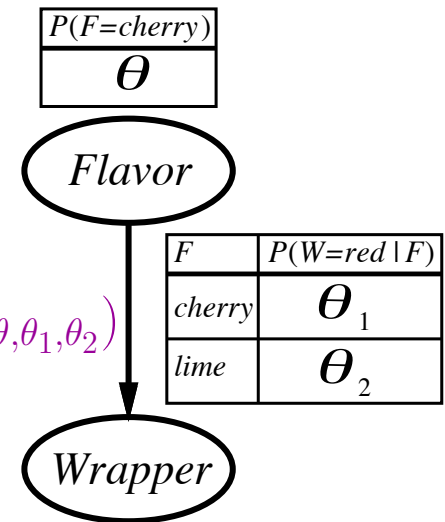
Likelihood for, e.g., cherry candy in green wrapper:

$$\begin{aligned}
 P(F = \text{cherry}, W = \text{green} | h_{\theta, \theta_1, \theta_2}) \\
 &= P(F = \text{cherry} | h_{\theta, \theta_1, \theta_2}) P(W = \text{green} | F = \text{cherry}, h_{\theta, \theta_1, \theta_2}) \\
 &= \theta \cdot (1 - \theta_1)
 \end{aligned}$$

N candies, r_c red-wrapped cherry candies, etc.:

$$P(\mathbf{d} | h_{\theta, \theta_1, \theta_2}) = \theta^c (1 - \theta)^\ell \cdot \theta_1^{r_c} (1 - \theta_1)^{g_c} \cdot \theta_2^{r_\ell} (1 - \theta_2)^{g_\ell}$$

$$\begin{aligned}
 L &= [c \log \theta + \ell \log(1 - \theta)] \\
 &\quad + [r_c \log \theta_1 + g_c \log(1 - \theta_1)] \\
 &\quad + [r_\ell \log \theta_2 + g_\ell \log(1 - \theta_2)]
 \end{aligned}$$



Multiple parameters contd.

Derivatives of L contain only the relevant parameter:

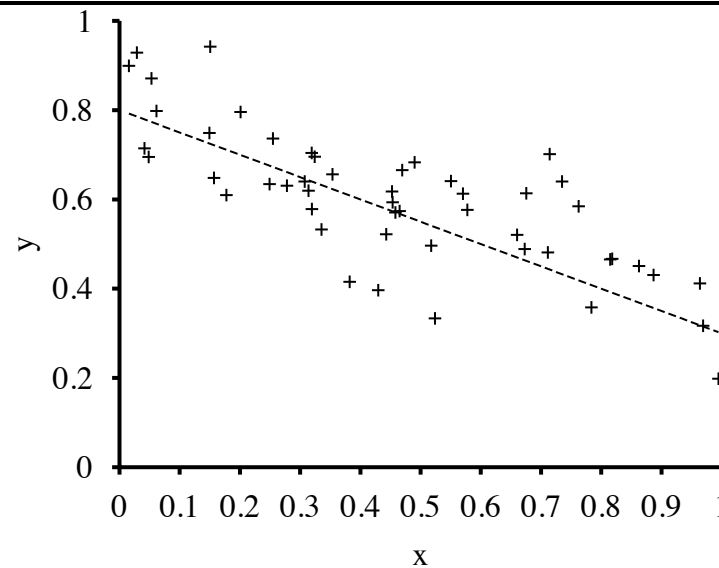
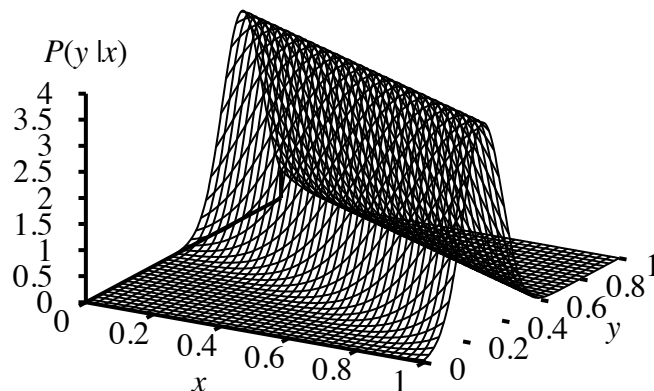
$$\frac{\partial L}{\partial \theta} = \frac{c}{\theta} - \frac{\ell}{1 - \theta} = 0 \quad \Rightarrow \quad \theta = \frac{c}{c + \ell}$$

$$\frac{\partial L}{\partial \theta_1} = \frac{r_c}{\theta_1} - \frac{g_c}{1 - \theta_1} = 0 \quad \Rightarrow \quad \theta_1 = \frac{r_c}{r_c + g_c}$$

$$\frac{\partial L}{\partial \theta_2} = \frac{r_\ell}{\theta_2} - \frac{g_\ell}{1 - \theta_2} = 0 \quad \Rightarrow \quad \theta_2 = \frac{r_\ell}{r_\ell + g_\ell}$$

With complete data, parameters can be learned separately

Example: linear Gaussian model



Maximizing $P(y|x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y-(\theta_1 x + \theta_2))^2}{2\sigma^2}}$ w.r.t. θ_1, θ_2

= minimizing $E = \sum_{j=1}^N (y_j - (\theta_1 x_j + \theta_2))^2$

That is, minimizing the sum of squared errors gives the ML solution for a linear fit **assuming Gaussian noise of fixed variance**

Continuous models use the same methods and principles as for the discrete case. The maths is trickier, though

Summary

Full Bayesian learning gives best possible predictions but is intractable

MAP learning balances complexity with accuracy on training data

Maximum likelihood assumes uniform prior, OK for large data sets

1. Choose a parameterized family of models to describe the data
requires substantial insight and sometimes new models
2. Write down the likelihood of the data as a function of the parameters
may require summing over hidden variables, i.e., inference
3. Write down the derivative of the log likelihood w.r.t. each parameter
4. Find the parameter values such that the derivatives are zero
may be hard/impossible; modern optimization techniques help