CSE 537 Fall 2015

LEARNING PROBABILISTIC MODELS

AIMA CHAPTER 20

Instructor: Sael Lee

Materials form AIMA resources, “Learning with Maximum Likelihood” by Andrew W. Moore and “The EM Algorithm: short tutorial” by S. Borman

OUTLINE

Agents can handle uncertainty by using the methods of probability and decision theory, but first they must learn their probabilistic theories of the world from experience by formulating the learning task itself as a process of probabilistic inference.

- Statistical learning
  - Bayesian learning
- Learning with Complete data
  - Maximum-likelihood parameter learning
- Learning with Hidden Variables: EM
  - General Form of EM
  - Unsupervised clustering: mixture of Gaussians
  - Learning Bayesian net with hidden variables
  - Learning HMM

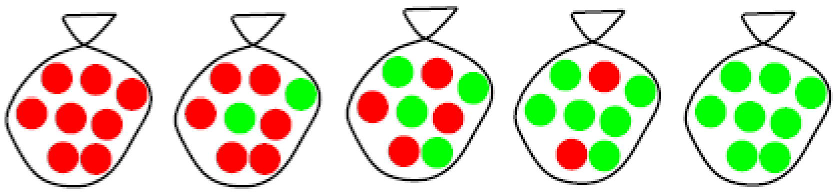
STATISTICAL LEARNING

- Bayesian view of learning:
  - Provides general solutions to the problems of noise, over-fitting and optimal prediction.
  -
- The **data** are **evidence**: instantiation of some or all of the random variables describing the domain.
- The **hypotheses** are probabilistic theories of how the domain works, including logical theories as a special case.

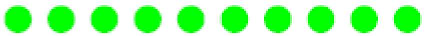
SURPRISE CANDY EXAMPLE

Suppose there are five kinds of bags of candies:

- 10% are *h1*: 100% cherry candies
- 20% are *h2*: 75% cherry candies + 25% lime candies
- 40% are *h3*: 50% cherry candies + 50% lime candies
- 20% are *h4*: 25% cherry candies + 75% lime candies
- 10% are *h5*: 100% lime candies



Given a new bag of candy, and we observe candies drawn from the bag:



TASK1: What kind of bag is it?
TASK2: What flavor will the next candy be?

POSTERIOR PROBABILITY OF HYPOTHESES

TASK1: What kind of bag is it? Let hypothesis H={h1,..,h5} denote the type of the bag.

Bayesian learning

Let **D** represent all the data with observed value **d**. Calculate the probability of each hypothesis given the data and predict on that basis.

Probabilities of each hypothesis are obtained by Bayes' rule.

likelihood

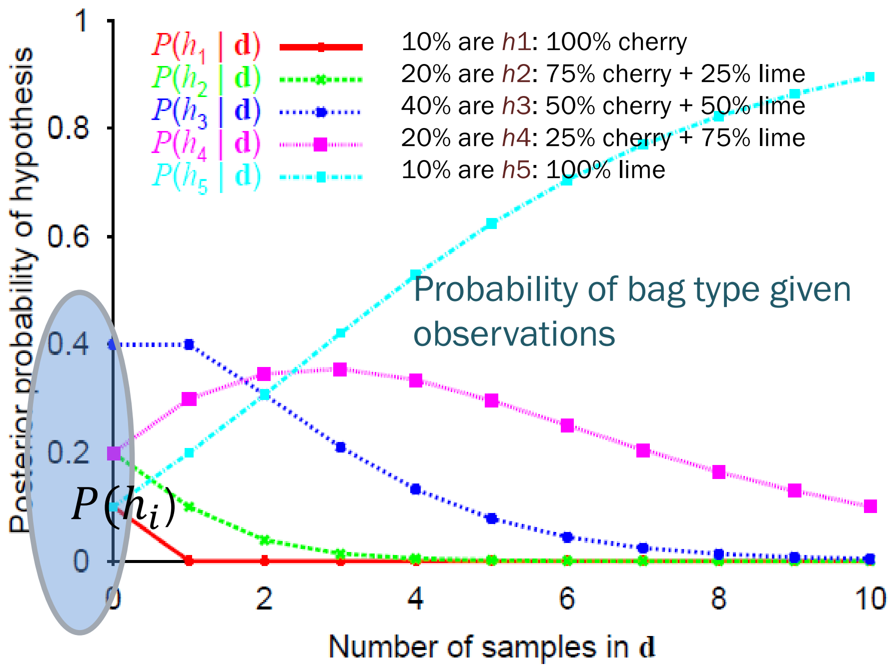
$$P(h_i|\mathbf{d}) = \alpha P(\mathbf{d}|h_i)P(h_i)$$

posterior

Hypothesis prior

Likelihood of data under i.i.d. assumption

$$P(d|h_i) = \prod_j P(d_j|h_i)$$



PREDICTION PROBABILITY

TASK2: What flavor will the next candy be?

Prediction about an unknown quantity X,

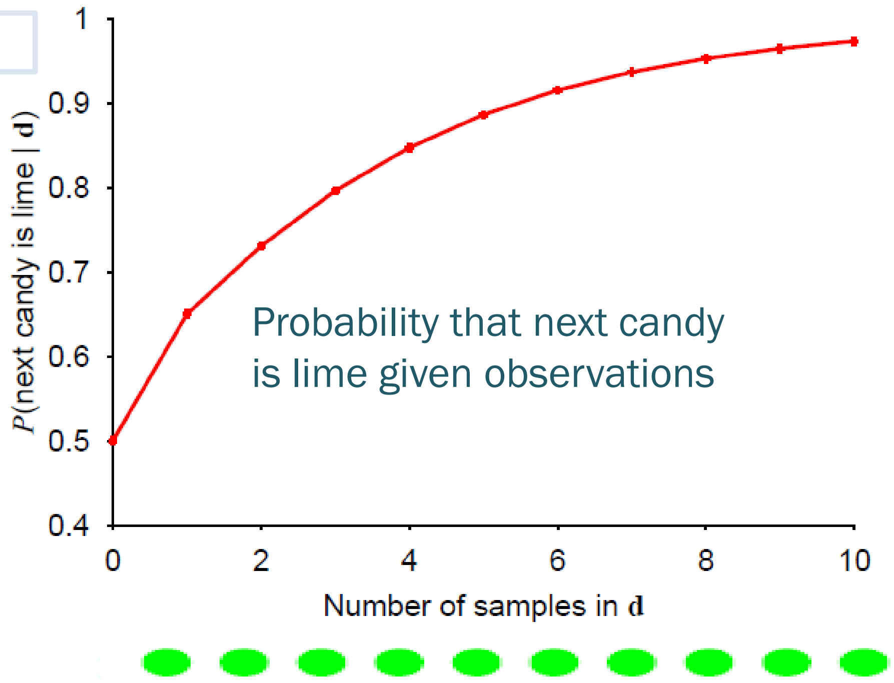
Predictions are weighted avg. over the predictions of the individual hypothesis.

posterior

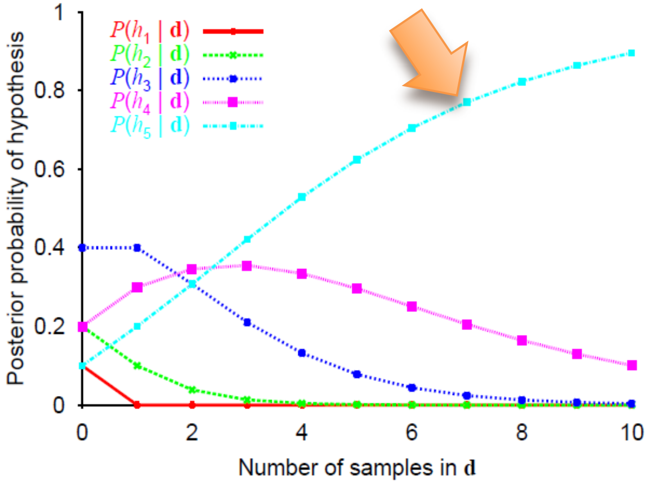
Prediction

$$P(X \mid d) = \sum_i P(X \mid d, h_i) P(h_i \mid d)$$
$$= \sum_i P(X \mid h_i) P(h_i \mid d)$$

assuming that each hypothesis determines a probability distribution over X.



OPTIMALITY OF BAYESIAN PREDICTION



- The Bayesian prediction eventually agrees with the true hypothesis
- For **any fixed prior** that does not rule out the true hypothesis, the posterior probability of any false hypothesis will, under certain technical conditions, eventually vanish.
- Bayesian prediction is **optimal** whether the data set be small or large. Given the hypothesis prior, any other prediction is expected to be correct less often.

REALITY

- In real learning problems, the hypothesis space is usually very large or infinite

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

Summing over the hypothesis space is often intractable

(e.g., 18,446,744,073,709,551,616 Boolean functions of 6 attributes)

Need approximation/simplified method for selecting the hypothesis

MAXIMUM A POSTERIORI (MAP) APPROXIMATION

Make predictions based on a single most probable hypothesis

$$P(X \mid d) \approx P(X \mid h_{MAP})$$

$$P(h_{MAP}) = \operatorname*{argmax}_{h_i}(P(h_i \mid d))$$

$$P(h_{MAP}) = \operatorname*{argmax}_{h_i}(P(d \mid h_i)P(h_i))$$

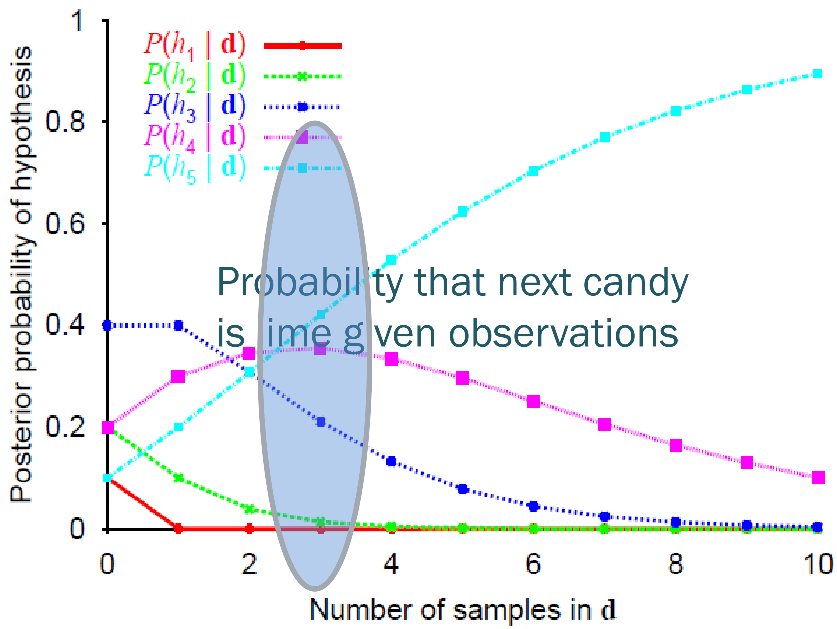
$$= \operatorname*{argmax}_{h_i}(\log(P(d \mid h_i)) + \log(P(h_i))$$

- MAP learning chooses the hypothesis that provides maximum *compression* of the data.
  - $\log_2 P(h_i)$: the number of bits required to specify the hypothesis h_i.
  - $\log_2 P(\mathbf{d} \mid h_i)$: the additional number of bits required to specify the data, given the hypothesis.

MAP VS BAYESIAN

EX> After three observations

MAP predict with probability 1 that next candy is lime
(pick h5)
Bayes will predict with probability 0.8 that net is lime



MAP & BAYESIAN – CONTROLLING COMPLEXITY

** BOTH MAP and Bayes penalize complexity using prior probability $P(h_i)$

- High $P(h_i)$ high penalty

Typically**, more complex hypothesis have a lower prior probability** – in part because there are casually many more complex hypothesis that simple hypotheses. On the other hand, more complex hypothesis save a greater capacity to fit the data.

MAXIMUM-LIKELIHOOD (ML) HYPOTHESIS APPROX.

Assume uniform prior over the space of hypothesis

MAP with uniform prior: Maximum-likelihood hypothesis

Becomes irrelevant if uniform

$$P(h_{MAP}) = \underset{h_i}{\operatorname{argmax}}(\log(P(d \mid h_i)) + \log(P(h_i)))$$

$$P(h_{ML}) = \underset{h_i}{\operatorname{argmax}}(\log(P(d \mid h_i)))$$

ML hypotheses is good for cases:

- Cannot trust the subjective nature of hypothesis prior
- No reason to prefer one hypothesis over another
    - When complexity of each hypothesis is all similar
- Good approximation when you have large dataset (problem if not)

LEARNING WITH COMPLETE DATA

The general task of learning a probability model, given data that are assumed to be generated form that model is called **density estimation.**

For simplicity, lets assume we have **complete data**, i.e., each data point contains values for every variable (feature) in the probability model being learned. – no missing data (fully observable)

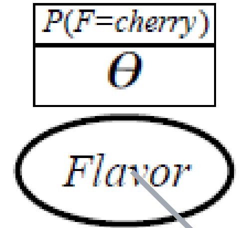
Parameter learning:
Finding the numerical parameters for a probability model whose structure if fixed.

Structure learning:
Finding the structure of the probability model.

ML PARAMETER LEARNING: DISCRETE VARIABLE

Parameter ranging form [0 .. 1]

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



Just one variable

Suppose we unwrap N candies, c cherries and $\ell = N - c$ limes
These are i.i.d. (independent, identically distributed) observations, so

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

<- Likelihood of observed data

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

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

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

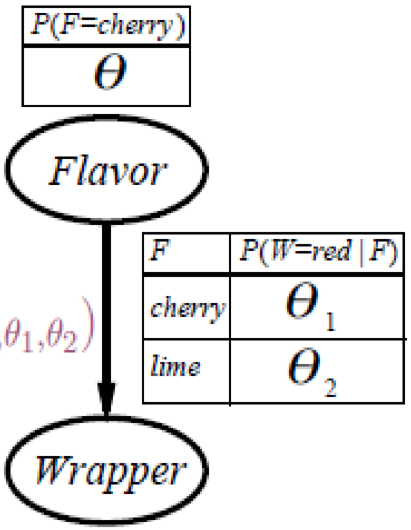
Seems sensible, but causes problems with 0 counts!

Finding maximum log likelihood

- ML parameter learning step:
  1. Write down an expression for the likelihood of the data as a function of parameters
  2. Write down the derivation of the log likelihood w.r.t. each parameters
  3. Find the parameter values such that the derivatives are zero
     - Non-trivial in practice
     - Use iterative methods and/or numerical optimization techniques
- Problem with ML
  - When the data set is small enough that some events have not yet been observed, the ML hypothesis assigns zero probability to those events.

ML: MULTIPLE PARAMETERS

Red/green wrapper depends probabilistically on flavor:



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

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

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

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

N candies unwrapped, c are cherries and *l* are limes

Take logarithm

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

With complete data, the ML parameter learning problem for a Bayesian network <u>decomposes into separate learning problems</u>, one for each parameter

ML: MULTIPLE PARAMETERS CONT.

Derivatives of L contain only the relevant parameter:

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

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

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

With complete data, parameters can be learned separately

ML FOR CONTINUOUS MODELS

- Example: Linear Gaussian model
  - Learning the parameters of a Gaussian density function on a single variable.
  - Data are generated as follows: $P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$.
  - Let the observed values be $x_1, \ldots, x_N$. Then the log likelihood is:

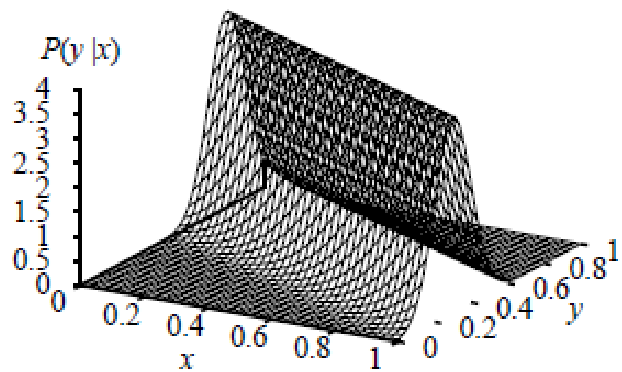
$$L = \sum_{j=1}^{N} \log \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_j-\mu)^2}{2\sigma^2}} = N(-\log\sqrt{2\pi} - \log\sigma) - \sum_{j=1}^{N} \frac{(x_j-\mu)^2}{2\sigma^2} .$$

  - Setting the derivatives to zero as usual, we obtain

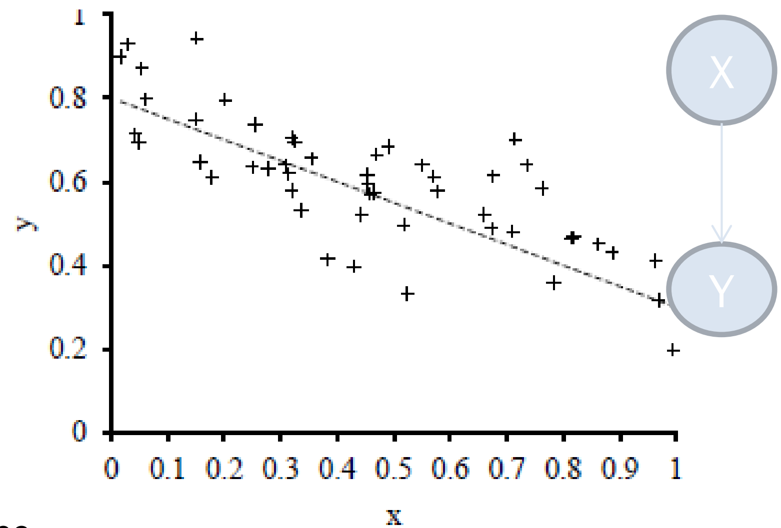
$$\begin{aligned}
\frac{\partial L}{\partial \mu} &= -\frac{1}{\sigma^2}\sum_{j=1}^{N}(x_j-\mu) = 0 & \Rightarrow \quad \mu &= \frac{\sum_j x_j}{N} \\
\frac{\partial L}{\partial \sigma} &= -\frac{N}{\sigma} + \frac{1}{\sigma^3}\sum_{j=1}^{N}(x_j-\mu)^2 = 0 & \Rightarrow \quad \sigma &= \sqrt{\frac{\sum_j (x_j-\mu)^2}{N}} .
\end{aligned}$$

ML FOR CONTINUOUS MODELS
EXAMPLE: LINEAR GAUSSIAN MODEL

EX> One continuous parent X an a continuous child Y. Y has Gaussian distribution whose <u>mean depends linearly</u> on the value of X and whose <u>std is fixed</u>.



linear Gaussian model described as
$y = \theta_1 x + \theta_2$ plus Gaussian noise with fixed variance.



A set of 50 data points generated from this 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 <u>assuming Gaussian noise of fixed variance</u>

BAYESIAN PARAMETER LEARNING

- Maximum-likelihood learning gives rise to some very simple procedures, but it has some serious deficiencies with small data sets
- The Bayesian approach to parameter learning:
  - Starts by defining a prior probability distribution (**hypothesis prior**) over the possible hypotheses.
  - Then, as data arrives, the posterior probability distribution is updated.