

Artificial Intelligence

V. Machine Learning

3. Statistical Learning

F. C. Langbein

School of Computer Science
and Informatics
Cardiff University



1.4

Overview

- Full Bayesian learning
- Maximum a posteriori learning
 - Minimum description length learning
- Maximum likelihood learning

F. C. Langbein, Artificial Intelligence – V. Machine Learning; 3. Statistical Learning

1

Full Bayesian Learning

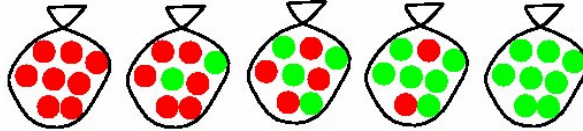
- View learning as Bayesian updating of a probability distribution over the *hypothesis space*
 - Hypothesis variable H with values $\{h_i\}$ and prior $P(H)$
 - j^{th} observation d_j gives random variable D_j
 - Training data $\mathbf{d} = [d_1, \dots, d_N]$
 - Given \mathbf{d} , each hypothesis has 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*
- Predict via likelihood-weighted average over hypotheses
$$P(X|\mathbf{d}) = \sum_i P(X|\mathbf{d}, h_i)P(h_i|\mathbf{d}) = \sum_i P(X|h_i)P(h_i|\mathbf{d})$$
- No need to pick one best-guess hypothesis

F. C. Langbein, Artificial Intelligence – V. Machine Learning; 3. Statistical Learning

2

Example

- 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



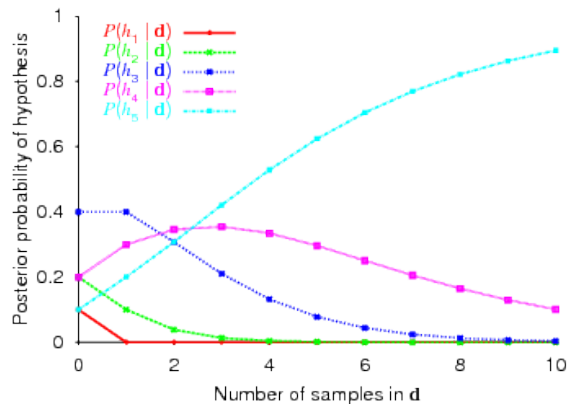
- Then we observe candies drawn from some bag:



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

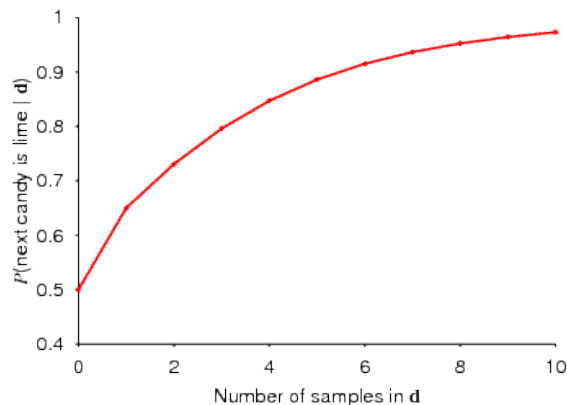
Posterior Probability of Hypotheses

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



Prediction Probability

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



MAP Approximation

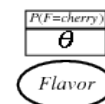
- Summing over the hypothesis space is often intractable (e.g., 18,446,744,073,709,551,616 Boolean functions of 6 attributes)
- *Maximum a posteriori* (MAP) learning:
 - choose h_{MAP} maximizing $P(h_i|\mathbf{d})$
 - I.e. maximise $P(\mathbf{d}|h_i)P(h_i)$ or $-\log P(\mathbf{d}|h_i) - \log P(h_i)$
- Log terms can be viewed as
 - bits to encode data given hypothesis +
 - bits to encode hypothesis
- Basic idea of *minimum description length* (MDL) learning
- For deterministic hypotheses,
 - $P(\mathbf{d}|h_i)$ is 1 if consistent, 0 otherwise
- MAP = simplest consistent hypothesis (cf. science)

ML Approximation

- For *large* data sets, prior becomes irrelevant
- *Maximum likelihood* (ML) learning:
 - choose h_{ML} maximising $P(\mathbf{d}|h_i)$
 - I.e. simply get the best fit to the data
 - Identical to MAP for uniform prior (reasonable if all hypotheses are of the same complexity)
- ML is the “standard” (non-Bayesian) statistical learning method
 - Initially no hypothesis is preferred (distrust any subjective priors)

ML Parameter Learning in Bayes Net

- Bag from a new manufacturer:
 - What is the 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



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

ML Parameter Learning in Bayes Net

- Maximise this w.r.t. θ — easier for the *log-likelihood*:

$$\begin{aligned} L(\mathbf{d}|\mathbf{h}_\theta) &= \log P(\mathbf{d}|\mathbf{h}_\theta) = \sum_{j=1}^N \log P(\mathbf{d}_j|\mathbf{h}_\theta) \\ &= c \log \theta + \ell \log(1 - \theta) \end{aligned}$$

Maximise:

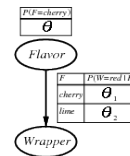
$$\begin{aligned} \frac{dL(\mathbf{d}|\mathbf{h}_\theta)}{d\theta} &= \frac{c}{\theta} - \frac{\ell}{1-\theta} = 0 \\ \implies \theta &= \frac{c}{c+\ell} = \frac{c}{N} \end{aligned}$$

- Seems sensible, but causes problems with 0 counts!

Multiple Parameters Example

- Red/green wrapper depends probabilistically on flavor
- Likelihood for, e.g., cherry candy in green wrapper:

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



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

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

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

Multiple Parameters Example

- Derivatives of L contain only the relevant parameter:

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

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

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

- With *complete data*, parameters can be learned separately

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 parameterised 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 optimisation techniques help