Predictions from Training Examples

We want to predict the output Y of a new case that has input $X = x$ given the training examples E:

$$
\begin{aligned}
p(Y|x \wedge E) &= \sum_{m \in Models} P(Y \wedge m|x \wedge E) \\
&= \sum_{m \in M} P(Y|m \wedge x \wedge E)P(m|x \wedge E) \\
&= \sum_{m \in M} P(Y|m \wedge x)P(m|E)
\end{aligned}
$$

Models is a set of mutually exclusive and covering hypotheses.

Learning Under Uncertainty

- We want to learn models from examples.

$$P(model|E) = \frac{P(E|model) \times P(model)}{P(E).}$$

- The likelihood, $P(E|model)$, is the probability that this model would have produced examples E.
- The prior, $P(model)$, encodes the learning bias

Bayesian Leaning of Probabilities

- Suppose there are two outcomes A and $\neg A$. We would like to learn the probability of A given some training examples, E.
- We can treat the probability of A as a real-valued random variable on the interval $[0, 1]$, called $probA$.

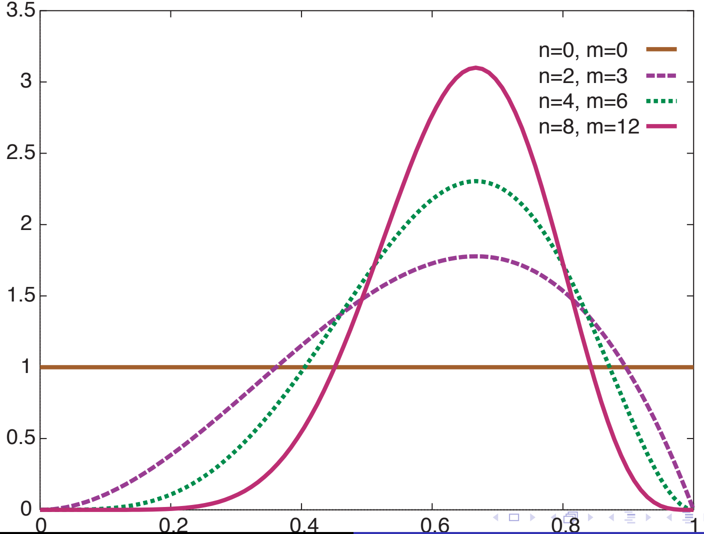
$$P(probA{=}p|E) = \frac{P(E|probA{=}p) \times P(probA{=}p)}{P(E)}$$

- Suppose the examples, E, is a sequence of n A's out of independent m trials,

$$P(E|probA{=}p) = p^n \times (1-p)^{m-n}$$

- Uniform prior: $P(probA{=}p) = 1$ for all $p \in [0, 1]$.

Posterior Probabilities for Different Training Examples



MAP model

- The maximum a posteriori probability (MAP) model is the model that maximizes $P(model|E)$. That is, it maximizes:

$$P(E|model) \times P(model)$$

- Thus it minimizes:

$$(-\log P(E|model)) + (-\log P(model))$$

which is the number of bits to send the examples, E, given the model plus the number of bits to send the model.

Information theory overview

- A bit is a binary digit.
- 1 bit can distinguish 2 items
- k bits can distinguish 2^k items
- n items can be distinguished using $\log_2 n$ bits
- Can you do better?

Information and Probability

Let's design a code to distinguish elements of $\{a, b, c, d\}$ with

$$P(a) = \frac{1}{2}, P(b) = \frac{1}{4}, P(c) = \frac{1}{8}, P(d) = \frac{1}{8}$$

Consider the code:

a	0	b	10	c	110	d	111

This code sometimes uses 1 bit and sometimes uses 3 bits. On average, it uses

$$\begin{aligned} & P(a) \times 1 + P(b) \times 2 + P(c) \times 3 + P(d) \times 3 \\ = \; & \frac{1}{2} + \frac{2}{4} + \frac{3}{8} + \frac{3}{8} = 1\frac{3}{4} \text{ bits.} \end{aligned}$$

The string *aacabbda* has code 00110010101110.

Information Content

- To identify x, you need $-\log_2 P(x)$ bits.
- If you have a distribution over a set and want to a identify a member, you need the expected number of bits:

$$\sum_x -P(x) \times \log_2 P(x).$$

  This is the information content or entropy of the distribution.
- The expected number of bits it takes to describe a distribution given evidence e:

$$I(e) = \sum_x -P(x|e) \times \log_2 P(x|e).$$

Information Gain

If you have a test that can distinguish the cases where α is true from the cases where α is false, the information gain from this test is:

$$I(true) - (P(\alpha) \times I(\alpha) + P(\neg\alpha) \times I(\neg\alpha)).$$

- $I(true)$ is the expected number of bits needed before the test
- $P(\alpha) \times I(\alpha) + P(\neg\alpha) \times I(\neg\alpha)$ is the expected number of bits after the test.

Averaging Over Models

- Idea: Rather than choosing the most likely model, average over all models, weighted by their posterior probabilities given the examples.
- If you have observed n A's out of m trials
  - the most likely value (MAP) is $\frac{n}{m}$
  - the expected value is $\frac{n+1}{m+2}$