# Outline

- Finish up Max likelihood learning of neural nets
- Bayes Optimal Classification
- Gibbs Sampling
- A practical use of Bayes Classifiers (text classification)
- Halving algorithm? EM? MDL? Autoclass?

# Most Probable Classification of New Instances

So far we've sought the most probable *hypothesis* given the data $D$ (i.e., $h_{MAP}$)

Given new instance $x$, what is its most probable *classification*?

- $h_{MAP}(x)$ is not the most probable classification!

Consider:

- Three possible hypotheses:

$$P(h_1|D) = .4,\ P(h_2|D) = .3,\ P(h_3|D) = .3$$

- Given new instance $x$,

$$h_1(x) = +,\ h_2(x) = -,\ h_3(x) = -$$

- What's most probable classification of $x$?

# Bayes Optimal Classifier

Assume target function $f : X \rightarrow V$ where $V$ finite

**Bayes optimal classification:** $\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$

Example:

$$
\begin{array}{lll}
P(h_1|D) = .4, & P(-|h_1) = 0, & P(+|h_1) = 1 \\
P(h_2|D) = .3, & P(-|h_2) = 1, & P(+|h_2) = 0 \\
P(h_3|D) = .3, & P(-|h_3) = 1, & P(+|h_3) = 0
\end{array}
$$

therefore

$$
\begin{aligned}
\sum_{h_i \in H} P(+|h_i)P(h_i|D) &= .4 \\
\sum_{h_i \in H} P(-|h_i)P(h_i|D) &= .6
\end{aligned}
$$

and

$$
\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) = -
$$

# Bayes Optimal Classifier

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

- No other method using the same hypothesis space can outperform the Bayes Optimal Classifier (in terms of expected accuracy)
- Notice Bayes Opt. Classifier doesn't necessarily correspond to any single hypothesis in $H$

# Gibbs Classifier

Bayes optimal classifier provides best result, but can be expensive if many hypotheses.

Gibbs algorithm:

1. Choose one hypothesis at random, according to $P(h|D)$
2. Use this to classify new instance

Surprising fact: Assume target concepts are drawn at random from $H$ according to priors on $H$. Then:

$$E[error_{Gibbs}] \leq 2E[error_{BayesOptimal}]$$

Suppose correct, uniform prior distribution over $H$, then

- Pick any hypothesis from VS, with uniform probability
- Its expected error no worse than twice Bayes optimal

# Minimum Description Length Principle

$$
\begin{aligned}
h_{MAP} &= \arg\max_{h \in H} P(D|h)P(h) \\
&= \arg\max_{h \in H} \log_2 P(D|h) + \log_2 P(h) \qquad (1)
\end{aligned}
$$

Interesting fact from information theory:

> The optimal (shortest expected coding length) code for an event with probability $p$ is $-\log_2 p$ bits.

So interpret (1):

- $\log_2 P(h)$ is length of $h$ under optimal code
- $\log_2 P(D|h)$ is length of $D$ given $h$ under optimal code (if assume $x_i$ fixed or indep of $h$, this is just length of encoding the misclassifications)

$\rightarrow$ prefer the hypothesis that minimizes

$$length(h) + length(misclassifications)$$

# Can Bayesian Methods be Practical?

NewsWeeder: [Lang, 95] Machine Learning Conference

- Nobody can read 9,000 newsgroups
- Allow user to rate articles they do read
- Learn to predict user ratings
- Fetch top articles from the other 8,994 newgroups

$\rightarrow$ Empirical Media Inc.

Bayesian/MDL approach

# One Bayesian Approach to Text Classification

Target concept: $Interesting? : Text \rightarrow \{+, -\}$

1. Re-represent each article by a vector of length 30,000
   - one feature per possible word
   - value is 1 if word occurs, 0 otherwise
2. Use training examples to learn
   - $P(Text\_vector|+)$
   - $P(+)$
   - $P(Text\_vector|-)$
   - $P(-)$

For new article, $text\_vector_{new}$ chose

$$c_{MAP} = \arg \max_{c \in \{+,-\}} P(text\_vector_{new}|c)P(c)$$

# Text Classification

How can we estimate $P(Text\_vector_{new}|+)$?

$P(Text\_vector) = P(word_1, word_2, \ldots, word_{30,000})$

Probably no training examples exactly match $Text\_vector_{new}$

Make assumption that words are independent:

- Assume $P(word_i, word_j) = P(word_i)P(word_j)$
- In fact, $P(word_i, word_j) = P(word_i|word_j)P(word_j)$

Obviously incorrect, but what else can we do??!??!? Now we have:

$$P(Text\_vector|+) = \prod_{i=1}^{|Text\_vector|} P(word_i|+)$$

# Bayesian Text Classification

Notice

- Assumed articles could be summarized by $text\_vector$
- Assumed independent probabilities of word occurance
- Final algorithm

  $P(w_i = 1|c) \leftarrow$ fraction of articles in $c$ containing word $i$

  $P(w_i = 0|c) \leftarrow (1 - P(w_i = 1|c))$

  $$Interesting?(text\_vec) = \arg\max_{c \in \{+,-\}} \prod_{i=1}^{|Text\_vec|} P(w_i|c)P(c)$$

- We're using Bayes rule here to select among target function *values*, not to select among *hypotheses* in $H = \{h | h : text\_vector \rightarrow \{+,-\}\}$

# NewsWeeder

NewsWeeder follows this approach, with some wrinkles:

- Predicts ratings 0 through 5 instead of +,-
- More subtle method for modeling word probabilities

  We suggested

  $P(w_i = 1|c) \leftarrow$ fraction of articles in $c$ containing word $i$

  but it considers *document length* as well:

  $P(w_i = 1|c, length) \leftarrow$ a complicated function of c, length
- Uses adaptive (MDL) method to select among models for $P(w_i = 1|c, length)$