Discrete Features I
Exploring Independence and Modeling

Matthew Stone
CS 520, Spring 2000
Lecture 7

Bayesian Decision Theory
DISCRETE FEATURES

- **Finite set of *c* states of nature**
- **Measurement is a discrete feature vector**
  - E.g., $\mathbf{x} \in \{0,1\}^k$
  - *k*-dimensional binary feature space
- **Provides setting for many key algorithms**
  - Markov models, belief nets, etc.

Bayes Decision Rule

- **Infer most likely state given measurement**
  - Using Bayes formula, here:

$$P(\omega_i \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid \omega_i)P(\omega_i)}{P(\mathbf{x})}$$

Bayes Decision Rule

- **Again we have "curse of dimensionality"**
  - Need $c2^k$ numbers to specify distribution
- **Worse –**
  - To estimate parameters $P(\mathbf{x} \mid \omega_i)$ with expected accuracy $1/\varepsilon$
  - Need $c\varepsilon^2 2^k$ training samples

Possible Solution: Modeling

- **Provide a specification outlining sparse relationships among features and classes**

Model Zero Naïve Bayes Classification

- **Features are independent given the class**
- **"Model" requires *ck* parameters:**
  - Likelihoods $p_{ij} := P(x_j = 1 | \omega_i)$
- **We get**

$$P(\mathbf{x} | \omega_i) = \prod_{j=1}^{k} p_{ij}^{x_j} (1 - p_{ij})^{1 - x_j}$$

Naïve Bayes Classification
CONTINUED

- **Use usual discriminant function**

$$g_i(\mathbf{x}) = \ln P(\omega_i) + \ln P(\mathbf{x} \mid \omega_i)$$

– i.e:

$$g_i(\mathbf{x}) = \ln P(\omega_i) + \sum_{j=1}^{k} \left[x_j \ln p_{ij} + (1 - x_j) \ln(1 - p_{ij}) \right]$$

– i.e:

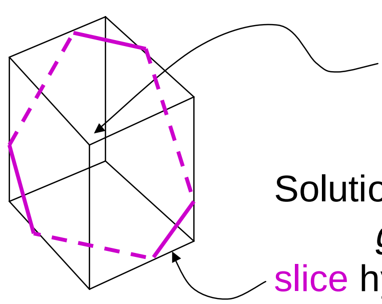
$$g_i(\mathbf{x}) = \mathbf{w}_i^{\mathrm{T}} \mathbf{x} + \theta$$

$$w_{ij} = \ln p_{ij} - \ln(1 - p_{ij})$$

$$\theta = \ln P(\omega_i) + \sum_{i=1}^{k} \ln(1 - p_{ij})$$

Visualization

- **Decision surfaces are hyperplanes**



Measurements occupy vertices of a hypercube

Solutions to equations

$$g_h(\mathbf{x}) = g_j(\mathbf{x})$$

slice hypercube

Case Study

- **Text classification**
  - Assign a natural language document to a predefined category based on content

Text Classification Examples

- **Index medical journal article**
- **Catalogue book for library**
- **Fit web page into *Yahoo!* Hierarchy**
- **Filter news feed for personal interest**
- **Automatically delete spam email**

Formalizing Text Classification

- **States of nature**
  - *c* states representing the different possible categories for documents, e.g.
    - *Email:* ω_1 – interesting
      ω_2 – spam

Formalizing Text Classification

- **Measurement x for any document**
  - *k*-component binary feature vector
  - Pick *k* useful English words
    - "useful" means occurs often with good correlation to some class
  - Set $x_i = 1$ if word *i* occurs in document

Sparse Data
(Aside)

- **English has tens of thousands of words**
  - But narrow to 1K that best discriminate
  - Still 10^{300} feature vectors
    - much larger than, e.g., *go* search space
    - no hope of describing P(**x**|c) without a model

Naïve Bayes Model

- **Assume features are independent**
  - Take maximum likelihood estimate for

$$p_{ij} := P(x_j = 1 \mid \omega_i)$$

  - That's just

$$\frac{\#\text{ of docs in class } \omega_i \text{ containing term } j}{\#\text{ of docs in class } \omega_i}$$

Naïve Bayes Model
(CONTINUED)

- **Given measurement x, Bayes formula has**

$$P(\omega_i \mid \mathbf{x}) = \frac{\left[\prod_j P(x_j = 1 \mid \omega_i)\right] P(\omega_i)}{P(\mathbf{x})}$$

- **So compute**

$$P(\omega_i \mid \mathbf{x}) \propto \frac{\#\text{ in class }\omega_i}{\#\text{ of docs}} \prod_j \frac{\#\text{ in class }\omega_i\text{ containing term }j}{\#\text{ in class }\omega_i}$$

Common Pitfall

- **With parameters set by MLE, you could easily end up with all posteriors zero**
- **To see how, suppose:**
  - Each feature occurs with some high probability *p* in a single class and some very low probability elsewhere: $1/\varepsilon$
  - You'd want some 2ε samples per class, but you can't get that many – only *n*
  - MLE estimate of $P(x_j = 1 \mid \omega_i)$ is often zero

Common Pitfall

- **With parameters set by MLE, you could easily end up with all posteriors zero**
- **To see how, suppose:**
  - The number k of features associated with each measurement is large
  - You expect a rare feature to occur on a test guy with probability roughly k/ε
  - Get rare feature not seen on any trainer from this category $-(k\varepsilon - n)/\varepsilon^2$ of the time

Sparse Data Requires Smoothing

- **Redistribute probability mass**
  - from what you saw
  - to what you didn't see
  - since you know other things can happen

Simple Smoothing: Deleted Estimation

- **Key question is often**
  - How often do you expect features in test data that never occur in training?
- **Deleted estimation finds this**
  - by splitting training data
  - and answering question empirically

Deleted Estimation

(CONTINUED)

- **Take first half**
  - N_0^1 – how many features don't occur there
  - C_0^{12} – how many of these occur in half two
- **Take second half**
  - N_0^2 – how many features don't occur there
  - C_0^{21} – how many of these occur in half two

Deleted Estimation
(CONTINUED)

- **This gives evidence about how often new things happen**

$$r_0 = \frac{C_0^{12} + C_0^{21}}{N_0^1 + N_0^2}$$

- **Smoothed value replaces MLE estimate**
  - Similar smoothed values required for other counts, to ensure probabilities sum to one