CS 188: Artificial Intelligence
Spring 2010

Lecture 23: Perceptrons
4/15/2010

Pieter Abbeel – UC Berkeley
Many slides adapted from Dan Klein.

Announcements

- Project 4: due tonight.
- W7: out tonight.
- Final Contest: up and running!

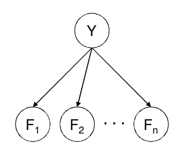
Outline

- Naïve Bayes recap
- Smoothing
- Generative vs. Discriminative
- Perceptron

Recap: General Naïve Bayes

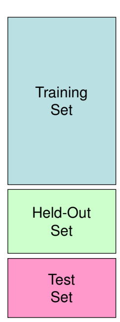
- A general *naïve Bayes* model:
  - Y: label to be predicted
  - $F_1, \dots, F_n$: features of each instance

$$P(Y, F_1 \dots F_n) = P(Y) \prod_i P(F_i|Y)$$



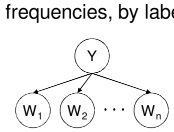
Naïve Bayes Training

- Data: labeled instances, e.g. emails marked as spam/ham by a person
  - Divide into training, held-out, and test
- Features are known for every training, held-out and test instance
- Estimation: count feature values in the training set and normalize to get maximum likelihood estimates of probabilities
- Smoothing (aka regularization): adjust estimates to account for unseen data

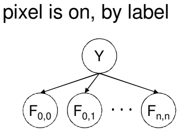


Example Naïve Bayes Models

- Bag-of-words for text
  - One feature for every word position in the document
  - All features share the same conditional distributions
  - Maximum likelihood estimates: word frequencies, by label



- Pixels for images
  - One feature for every pixel, indicating whether it is on (black)
  - Each pixel has a different conditional distribution
  - Maximum likelihood estimates: how often a pixel is on, by label



Outline

- Naïve Bayes recap
- *Smoothing*
- Generative vs. Discriminative
- Perceptron

Recap: Laplace Smoothing

- Laplace's estimate (extended):
  - Pretend you saw every outcome k extra times

$$P_{LAP,k}(x) = \frac{c(x)+k}{c(\cdot)+k|X|}$$

  - What's Laplace with k = 0?
  - k is the strength of the prior

H H T

$$P_{LAP,0}(X) = \left\langle \frac{2}{3}, \frac{1}{3} \right\rangle$$

$$P_{LAP,1}(X) = \left\langle \frac{3}{5}, \frac{2}{5} \right\rangle$$

$$P_{LAP,100}(X) = \left\langle \frac{102}{203}, \frac{101}{203} \right\rangle$$

- Laplace for conditionals:
  - Smooth each condition:
  - Can be derived by dividing

$$P_{LAP,k}(x|y) = \frac{c(x,y)+k}{c(\cdot,y)+k|X|}$$

Better: Linear Interpolation

- Linear interpolation for conditional likelihoods
  - **Idea**: the conditional probability of a feature x given a label y should be close to the marginal probability of x
  - **Example**: A rare word like "interpolation" should be similarly rare in both ham and spam (a priori)
  - **Procedure**: Collect relative frequency estimates of both conditional and marginal, then average

$$P_{ML}(x|y) = \frac{\text{count}(x,y)}{\text{count}(\cdot,y)} \qquad P_{ML}(x) = \frac{\text{count}(x)}{\text{count}(\cdot)}$$

$$P_{LIN}(x|y) = (1-\alpha)P_{ML}(x|y) + (\alpha)P_{ML}(x)$$

  - **Effect**: Features have odds ratios closer to 1

Real NB: Smoothing

- Odds ratios without smoothing:

$$\frac{P(W|\text{ham})}{P(W|\text{spam})}$$

```
south-west : inf
nation     : inf
morally    : inf
nicely     : inf
extent     : inf
...
```

$$\frac{P(W|\text{spam})}{P(W|\text{ham})}$$

```
screens    : inf
minute     : inf
guaranteed : inf
$205.00    : inf
delivery   : inf
...
```

Real NB: Smoothing

- Odds ratios after smoothing:

$$\frac{P(W|\text{ham})}{P(W|\text{spam})}$$

```
helvetica : 11.4
seems     : 10.8
group     : 10.2
ago       :  8.4
areas     :  8.3
...
```

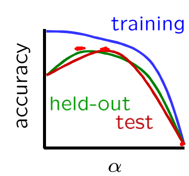
$$\frac{P(W|\text{spam})}{P(W|\text{ham})}$$

```
verdana : 28.8
Credit  : 28.4
ORDER   : 27.2
<FONT>  : 26.9
money   : 26.5
...
```

Do these make more sense?

Tuning on Held-Out Data

- Now we've got two kinds of unknowns
  - Parameters: $P(F_i|Y)$ and P(Y)
  - Hyperparameters, like the amount of smoothing to do: k, α
- Where to learn which unknowns
  - Learn parameters from training set
  - Can't tune hyperparameters on training data (why?)
  - For each possible value of the hyperparameters, train and test on the held-out data
  - Choose the best value and do a final test on the test data



Proportion of $P_{ML}(x)$ in P(x|y)

Baselines

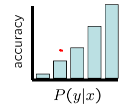
- First task when classifying: get a baseline
  - Baselines are very simple "straw man" procedures
  - Help determine how hard the task is
  - Help know what a "good" accuracy is
- Weak baseline: most frequent label classifier
  - Gives all test instances whatever label was most common in the training set
  - E.g. for spam filtering, might label everything as spam
  - Accuracy might be very high if the problem is skewed
- When conducting real research, we usually use previous work as a (strong) baseline

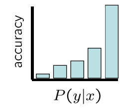
Confidences from a Classifier

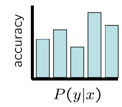
- The confidence of a classifier:
  - Posterior of the most likely label

$$\text{confidence}(x) = \max_y P(y|x)$$

  - Represents how sure the classifier is of the classification
  - Any probabilistic model will have confidences
  - No guarantee confidence is correct
- Calibration
  - Strong calibration: confidence predicts accuracy rate
  - Weak calibration: higher confidences mean higher accuracy
  - What's the value of calibration?







Naïve Bayes Summary

- Bayes rule lets us do diagnostic queries with causal probabilities
- The naïve Bayes assumption takes all features to be independent given the class label
- We can build classifiers out of a naïve Bayes model using training data
- Smoothing estimates is important in real systems
- Confidences are useful when the classifier is calibrated

What to Do About Errors

- Problem: there's still spam in your inbox
- Need more features – words aren't enough!
  - Have you emailed the sender before?
  - Have 1K other people just gotten the same email?
  - Is the sending information consistent?
  - Is the email in ALL CAPS?
  - Do inline URLs point where they say they point?
  - Does the email address you by (your) name?
- Naïve Bayes models can incorporate a variety of features, but tend to do best in homogeneous cases (e.g. all features are word occurrences)

17

Outline

- Naïve Bayes recap
- Smoothing
- *Generative vs. Discriminative*
- Perceptron

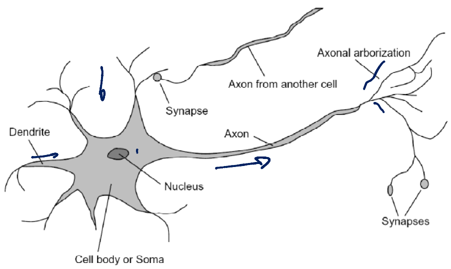
Generative vs. Discriminative

- Generative classifiers:
  - E.g. naïve Bayes
  - A causal model with evidence variables
  - Query model for causes given evidence
- Discriminative classifiers:
  - No causal model, no Bayes rule, often no probabilities at all!
  - Try to predict the label Y directly from X
  - Robust, accurate with varied features
  - Loosely: mistake driven rather than model driven

21

Some (Simplified) Biology

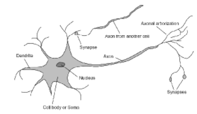
- Very loose inspiration: human neurons



22

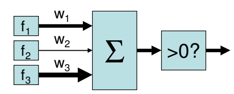
Linear Classifiers

- Inputs are feature values
- Each feature has a weight
- Sum is the activation



$$\text{activation}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$

- If the activation is:
  - Positive, output +1
  - Negative, output -1



23

Example: Spam

- Imagine 4 features (spam is "positive" class):
  - free (number of occurrences of "free")
  - money (occurrences of "money")
  - BIAS (intercept, always has value 1)

$w \cdot f(x)$

$\sum_i w_i \cdot f_i(x)$

x: "free money"

$f(x)$:

```
BIAS  :  1
free  :  1
money :  1
...
```

w:

```
BIAS  : -3
free  :  4
money :  2
...
```

$(1)(-3) +$
$(1)(4) +$
$(1)(2) +$
$\ldots$
$= 3$

Binary Decision Rule

- In the space of feature vectors
  - Examples are points
  - Any weight vector is a hyperplane
  - One side corresponds to Y=+1
  - Other corresponds to Y=-1

w

```
BIAS  : -3
free  :  4
money :  2
...
```