# NATURAL LANGUAGE LEARNING: LINEAR MODELS

David Kauchak
CS159, Spring 2011

## Admin

- Assignment 2
  - Perplexity
    - What was the best training set size?
  - In general, come talk to me if you're having problems
  - Real world: debugging is hard!
  - Java skills
- Assignment 3
- Assignment 4, out today

## The mind-reading game

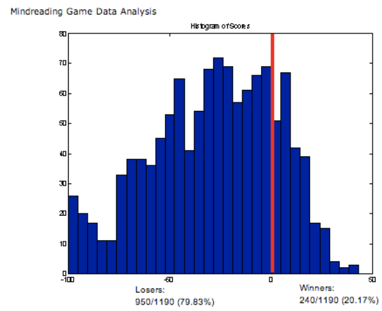
How good are you at guessing random numbers?

```
Repeat 100 times:
  Computer guesses whether you'll type 0/1
  You type 0 or 1
```

http://seed.ucsd.edu/~mindreader/
[written by Y. Freund and R. Schapire]

## The mind-reading game

The computer is right much more than half the time…



## The mind-reading game

The computer is right much more than half the time…

*Strategy*: computer predicts next keystroke based on the last few (maintains weights on different patterns)

There are patterns everywhere… even in "randomness"!

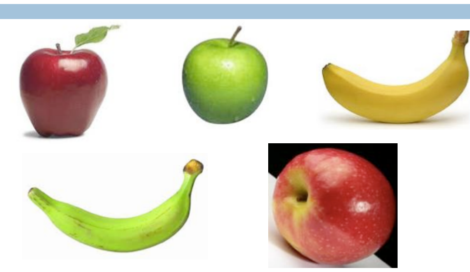
## Why machine learning?

- Lot's of data
- Hand-written rules just don't do it
- Performance is much better than what people can do
- Why not just study machine learning?
  - Domain knowledge/expertise is still very important
  - What types of features to use
  - What models are important

## Machine learning problems

- Lots of different types of problems
  - What data is available:
    - Supervised, unsupervised, semi-supervised, reinforcement learning
  - How are we getting the data:
    - online vs. offline learning
  - Type of model:
    - generative vs. disciminative
    - parametric vs. non-parametric
    - SVM, NB, decision tree, k-means
  - What are we trying to predict:
    - classification vs. regression

## Unsupervised learning

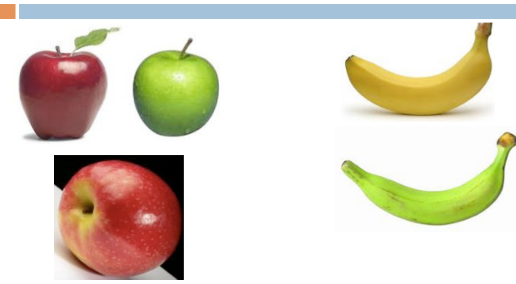


Unupervised learning: given data, but no labels

## Unsupervised learning

- Much easier to get our hands on unlabeled data
- EM was an unsupervised approach
  - learned clusters/groups without any label
  - learned grammar probabilities without trees
  - learned HMM probabilities without labels
- Because there is no label, often can get odd results
  - grammar learned by inside-outside often has little relation to linguistically motivated grammar
  - may cluster bananas/apples or green/red/yellow

## Supervised learning



**APPLES** **BANANAS**

Supervised learning: given labeled data

## Supervised learning

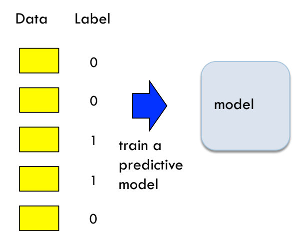
- Given labeled examples, learn to label unlabeled examples

**APPLE or BANANA?**

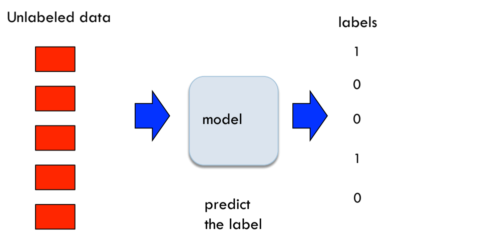
Supervised learning: learn to classify unlabeled

## Supervised learning: training

Labeled data

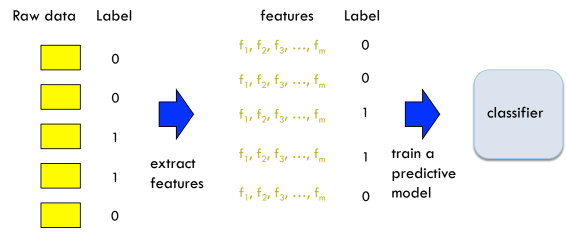


## Supervised learning: testing/classifying



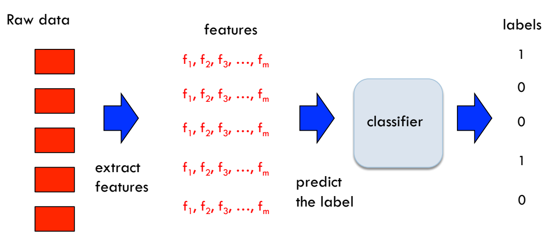

## Feature based learning

### Training or learning phase




## Feature based learning

### Testing or classification phase




## Feature examples

## Feature examples

Raw data

Features

Clinton said banana repeatedly last week on tv, "banana, banana, banana"

(1, 1, 1, 0, 0, 1, 0, 0, ...)

banana clinton said california across tv wrong capital

Occurrence of words

## Feature examples

Raw data

Features

Clinton said banana repeatedly last week on tv, "banana, banana, banana"

(4, 1, 1, 0, 0, 1, 0, 0, ...)

banana clinton said california across tv wrong capital

Frequency of word occurrence

## Feature examples

Raw data

Features

Clinton said banana repeatedly last week on tv, "banana, banana, banana"

(1, 1, 1, 0, 0, 1, 0, 0, ...)

banana repeatedly, clinton said, said banana, california schools, across the, tv banana, wrong way, capital city

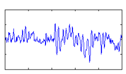
Occurrence of bigrams

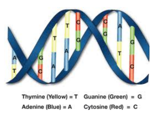
## Lots of other features

- POS: occurrence, counts, sequence
- Constituents
- Whether 'V1agra' occurred 15 times
- Whether 'banana' occurred more times than 'apple'
- If the document has a number in it
- …
- Features are very important, but we're going to focus on the models today

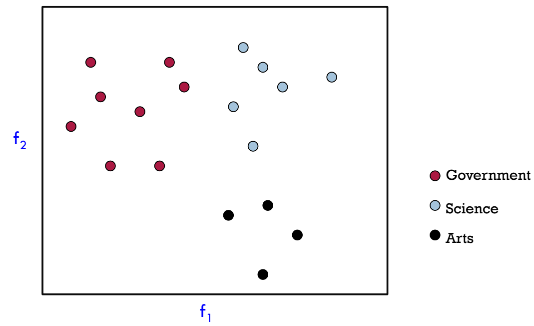
## Power of feature-base methods

- General purpose: any domain where we can represent a data point as a set of features, we can use the method

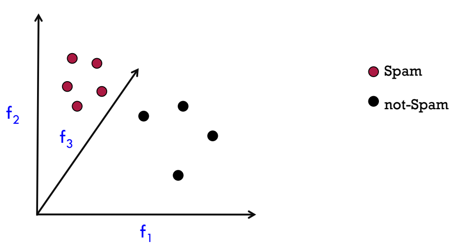




## The feature space




## The feature space




## Feature space

$f_1, f_2, f_3, \ldots, f_m$ m-dimensional space

How big will m be for us?

## Bayesian Classification

We represent a data item based on the features:

$$D = \langle f_1, f_2, \ldots, f_n \rangle$$

### Training

a: $p(a \mid D) = p(a \mid f_1, f_2, \ldots, f_n)$

b: $p(b \mid D) = p(b \mid f_1, f_2, \ldots, f_n)$

$$P(Label \mid f_1, f_2, \ldots, f_n)$$

For each label/class, **learn** a probability distribution based on the features

## Bayesian Classification

We represent a data item based on the features:

$$D = \langle f_1, f_2, \ldots, f_n \rangle$$

### Classifying

$$label = \underset{l \in Labels}{\operatorname{argmax}} P(l \mid f_1, f_2, \ldots, f_n)$$

Given an *new* example, classify it as the label with the largest conditional probability

## Bayes rule for classification

conditional (posterior) probability

prior probability

$$P(Label \mid Data) = \frac{P(D \mid C)P(C)}{P(D)}$$

Why not model P(Label|Data) directly?

## Bayesian classifiers

$$label = \underset{l \in Labels}{\operatorname{argmax}} P(l \mid f_1, f_2, \ldots, f_n)$$

different distributions for different labels

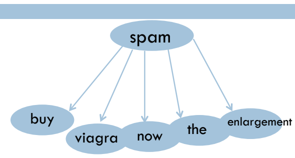
$$= \underset{l \in Labels}{\operatorname{argmax}} \frac{P(f_1, f_2, \ldots, f_n \mid l)P(l)}{P(f_1, f_2, \ldots, f_n)}$$

Bayes rule

$$= \underset{l \in Labels}{\operatorname{argmax}} P(f_1, f_2, \ldots, f_n \mid l)P(l)$$

two models to learn for each label/class

## The Naive Bayes Classifier




assume binary features for now

**Conditional Independence Assumption:** features are independent of each other given the class:

$$P(x_1,\ldots,x_n \mid l) = P(x_1 \mid l)P(x_2 \mid l)\cdots P(x_n \mid l)$$

$$label = \underset{l \in Labels}{\operatorname{argmax}} P(f_1 \mid l)P(f_2 \mid l)\ldots p(f_n \mid l)P(l)$$

## Estimating parameters

- p('v1agra' | spam)
- p('the' | spam)
- p('enlargement' | not-spam)
- …

For us:

$$label = \underset{l \in Labels}{\operatorname{argmax}} P(f_1 \mid l)P(f_2 \mid l)\ldots p(f_n \mid l)P(l)$$

## Maximum likelihood estimates

$$\hat{P}(l) = \frac{N(l)}{N}$$

$$\frac{\text{number of items with label}}{\text{total number of items}}$$

$$\hat{P}(f_i \mid l) = \frac{N(f_i, l)}{N(l)}$$

$$\frac{\text{number of items with the label with feature}}{\text{number of items with label}}$$

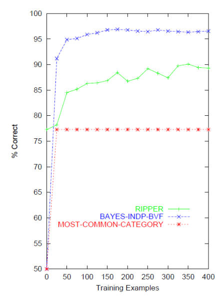
## Naïve Bayes Text Classification

- Features: word occurring in a document (though others could be used…)

$$label = \underset{l \in Labels}{\operatorname{argmax}} P(word_1 \mid l)P(word_2 \mid l)\ldots p(word_n \mid l)P(l)$$

- Does the Naïve Bayes assumption hold?
  - Are word occurrences independent given the label?
- Lot's of text classification problems
  - sentiment analysis: positive vs. negative reviews
  - category classification
  - spam

## Naive Bayes on spam email



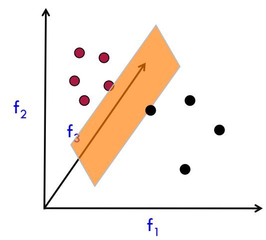

http://www.cnbc.cmu.edu/~jp/research/email.paper.pdf

## Linear models

- A linear model predicts the label based on a weighted, linear combination of the features

$$prediction = w_0 + w_1 f_1 + w_2 f_2 + \ldots + w_m f_m$$

- For two classes, a linear model can be viewed as a plane (hyperplane) in the feature space

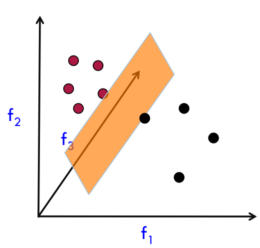



## Linear models

$$label = \underset{l \in Labels}{\operatorname{argmax}} P(f_1 \mid l) P(f_2 \mid l) \ldots p(f_n \mid l) P(l)$$

Is naïve bayes a linear model?

$$prediction = w_0 + w_1 f_1 + w_2 f_2 + \ldots + w_m f_m$$




## Linear models: NB

$$label = \underset{l \in Labels}{\operatorname{argmax}} P(f_1 \mid l) P(f_2 \mid l) \ldots p(f_n \mid l) P(l)$$

$$= \underset{l \in Labels}{\operatorname{argmax}} \log(P(f_1 \mid l) P(f_2 \mid l) \ldots p(f_n \mid l) P(l))$$

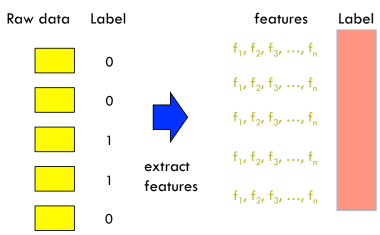
$$= \underset{l \in Labels}{\operatorname{argmax}} \log(P(f_1 \mid l)) + \log(P(f_2 \mid l)) + \ldots + \log(p(f_n \mid l)) + \log(P(l))$$

$$= \underset{l \in Labels}{\operatorname{argmax}} \log(P(f_1 \mid l)) + \log(P(f_2 \mid l)) + \ldots + \log(p(f_n \mid l)) + \log(P(l))$$

$$= \underset{l \in Labels}{\operatorname{argmax}} f_1 \log(P(f_1 \mid l)) + \bar{f}_1 \log(1 - P(f_1 \mid l)) + \ldots + \log(P(l))$$

$f_1 w_1$ $f_2 w_2$ $w_0$

## Regression vs. classification




classification: discrete (some finite set of labels)

regression: real value

## Regression vs. classification

| features | response |
| --- | --- |
| $f_1, f_2, f_3, \ldots, f_n$ | 1.0 |
| $f_1, f_2, f_3, \ldots, f_n$ | 2.3 |
| $f_1, f_2, f_3, \ldots, f_n$ | .3 |
| $f_1, f_2, f_3, \ldots, f_n$ | 100.4 |
| $f_1, f_2, f_3, \ldots, f_n$ | 123 |

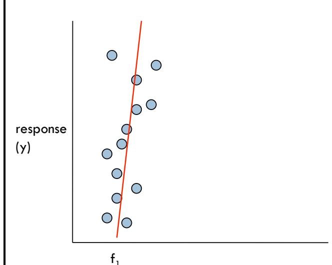
### Examples

- predict a readability score between 0-100 for a document
- predict the number of votes/reposts
- predict cost to insure
- predict income
- predict life longevity
- …

## Model-based regression

- Just like unsupervised approaches, may supervised approaches start with some model and try and "fit it" to the data
- Regression models
  - linear
  - logistic
  - polynomial
  - …

## Linear regression



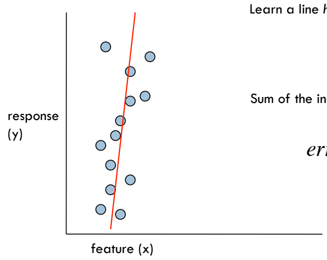

How can we find this line?

Given some points, find the ***line*** that best fits/explains the data

Our model is a line, i.e. we're assuming a linear relationship between the feature and the label value

$$h(y) = w_1 f_1 + w_0$$

## Linear regression




Learn a line *h* that minimizes some error function:

$$error(h) = ?$$

Sum of the individual errors:

$$error(h) = \sum_{i=1}^{n} \underbrace{|y_i - h(f_i)|}_{\text{for that example, what was the difference between actual and predicted}}$$

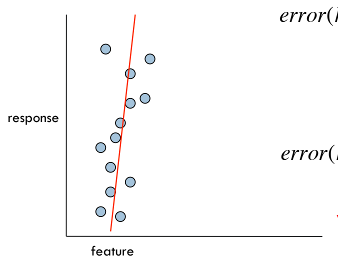
## Error minimization

- How do we find the minimum of an equation (think back to calculus…)?

$$error(h) = \sum_{i=1}^{n} |y_i - h(f_i)|$$

- Take the derivative, set to 0 and solve (going to be a min or a max)
- Any problems here?
- Ideas?

## Linear regression



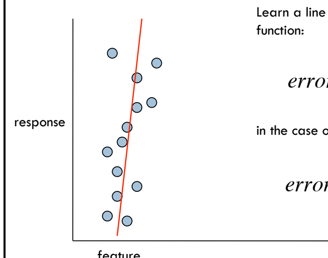

$$error(h) = \sum_{i=1}^{n} |y_i - h(f_i)|$$

$$\Downarrow$$

$$error(h) = \sum_{i=1}^{n} (y_i - h(f_i))^2$$

what's the difference?

## Linear regression




Learn a line *h* that minimizes an error function:

$$error(h) = \sum_{i=1}^{n} (y_i - h(f_i))^2$$

in the case of a 2d line:

$$error(h) = \sum_{i=1}^{n} (y_i - \underbrace{(w_1 f_i + w_0)}_{\text{function for a line}})^2$$

## Linear regression

- We'd like to *minimize* the error
  - Find $w_1$ and $w_0$ such that the error is minimized

$$error(h) = \sum_{i=1}^{n} (y_i - (w_1 f_i + w_0))^2$$

- We can solve this in closed form

## Multiple linear regression

- Often, we don't just have one feature, but have many features, say *m*
- Now we have a line in m dimensions
- Still just a line

$$h(\bar{f}) = w_0 + w_1 f_1 + w_2 f_2 + \ldots + w_m f_m$$

weights

A linear model is additive. The weight of the feature dimension specifies importance/direction

## Multiple linear regression

- We can still calculate the squared error like before

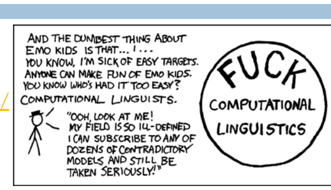
$$h(\bar{f}) = w_0 + w_1 f_1 + w_2 f_2 + \ldots + w_m f_m$$

$$error(h) = \sum_{i=1}^{n} (y_i - (w_0 + w_1 f_1 + w_2 f_2 + \ldots + w_m f_m))^2$$

Still can solve this exactly!

## Probabilistic classification

- We're NLP people
- We like probabilities!
- http://xkcd.com/114/




- We'd like to do something like regression, but that gives us a probability

## Classification

$$p(1 \mid x_1, x_2, \ldots, x_m) = w_0 + x_1 w_1 + w_2 x_2 + \ldots + w_m x_m$$

- Nothing constrains it to be a probability
- Could still have combination of features and weight that exceeds 1 or is below 0

## The challenge

$w_0 + x_1 w_1 + w_2 x_2 + \ldots + w_m x_m$ $\boxed{p(1 \mid x_1, x_2, \ldots, x_m)}$

Linear regression

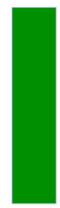
probability

$+\infty$

$-\infty$

Find some equation based on the probability that ranges from $-\infty$ to $+\infty$

1



0

## Odds ratio

- Rather than predict the probability, we can predict the ratio of 1/0 (true/false)
- Predict the **odds** that it is 1 (true): How much more likely is 1 than 0.
- Does this help us?

$$\frac{P(1 \mid x_1, x_2, \ldots, x_m)}{P(0 \mid x_1, x_2, \ldots, x_m)} = \frac{P(1 \mid x_1, x_2, \ldots, x_m)}{1 - P(1 \mid x_1, x_2, \ldots, x_m)} = w_0 + x_1 w_1 + w_2 x_2 + \ldots + w_m x_m$$

## Odds ratio

$w_0 + x_1 w_1 + w_2 x_2 + \ldots + w_m x_m$ $\dfrac{P(1 \mid x_1, x_2, \ldots, x_m)}{1 - P(1 \mid x_1, x_2, \ldots, x_m)}$

Linear regression

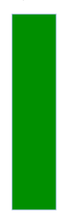
odds ratio

$+\infty$

$-\infty$

Where is the dividing line between class 1 and class 0 being selected?

$+\infty$



0

## Odds ratio

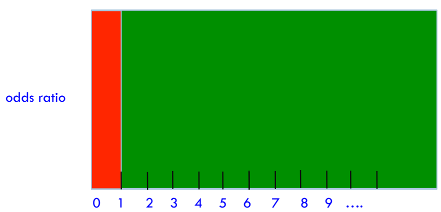
$P(1 \mid x_1, x_2, \ldots, x_m) > P(0 \mid x_1, x_2, \ldots, x_m)$

$P(1 \mid x_1, x_2, \ldots, x_m) > 1 - P(1 \mid x_1, x_2, \ldots, x_m)$

$$\frac{P(1 \mid x_1, x_2, \ldots, x_m)}{1 - P(1 \mid x_1, x_2, \ldots, x_m)}$$

We're trying to find some transformation that transforms the odds ratio to a number that is -∞ to +∞

Does this suggest another transformation?



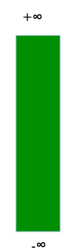

## Log odds (logit function)

$w_0 + x_1 w_1 + w_2 x_2 + \ldots + w_m x_m$

Linear regression


$$\log \frac{P(1 \mid x_1, x_2, \ldots, x_m)}{1 - P(1 \mid x_1, x_2, \ldots, x_m)}$$

odds ratio




How do we get the probability of an example?

## Log odds (logit function)

$$\log \frac{P(1 \mid x_1, x_2, \ldots, x_m)}{1 - P(1 \mid x_1, x_2, \ldots, x_m)} = w_0 + w_1 x_2 + w_2 x_2 + \ldots + w_m x_m$$

$$\frac{P(1 \mid x_1, x_2, \ldots, x_m)}{1 - P(1 \mid x_1, x_2, \ldots, x_m)} = e^{w_0 + w_1 x_2 + w_2 x_2 + \ldots + w_m x_m}$$

$$P(1 \mid x_1, x_2, \ldots, x_m) = 1 - P(1 \mid x_1, x_2, \ldots, x_m) e^{w_0 + w_1 x_2 + w_2 x_2 + \ldots + w_m x_m}$$

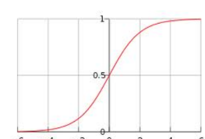
…

$$P(1 \mid x_1, x_2, \ldots, x_m) = \frac{1}{1 + e^{-(w_0 + w_1 x_2 + w_2 x_2 + \ldots + w_m x_m)}}$$

anyone recognize this?

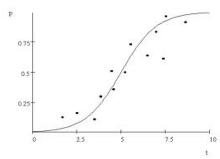
## Logistic function

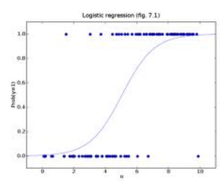
$$logistic = \frac{1}{1 + e^{-x}}$$



## Logistic regression

- Find the best fit of the data based on a logistic





## Logistic regression

- How would we classify examples once we had a trained model?

$$\log \frac{P(1 \mid x_1, x_2, \ldots, x_m)}{1 - P(1 \mid x_1, x_2, \ldots, x_m)} = w_0 + w_1 x_2 + w_2 x_2 + \ldots + w_m x_m$$

- If the sum > 0 then p(1)/p(0) > 1, so positive
- if the sum < 0 then p(1)/p(0) < 1, so negative
- Still a *linear* classifier (decision boundary is a line)