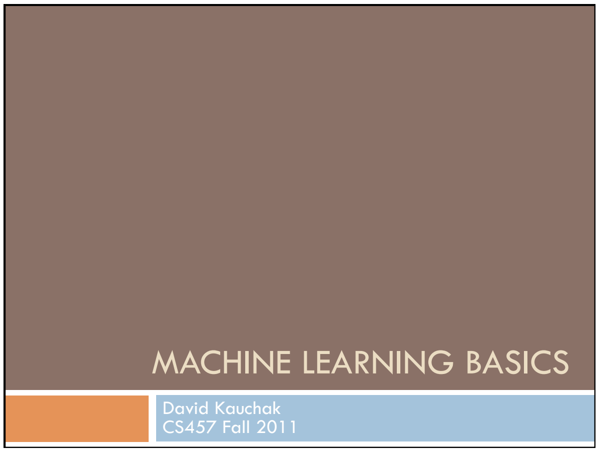
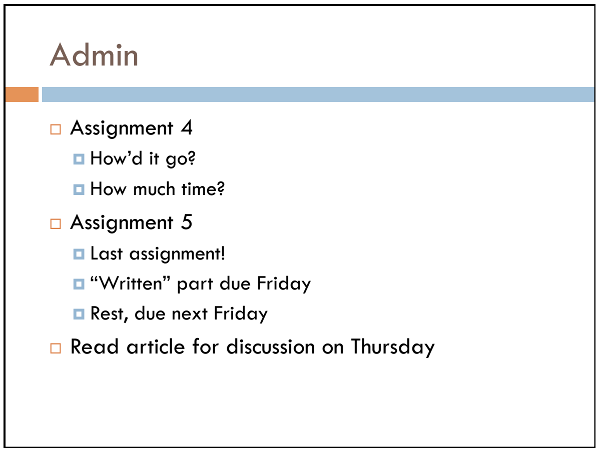


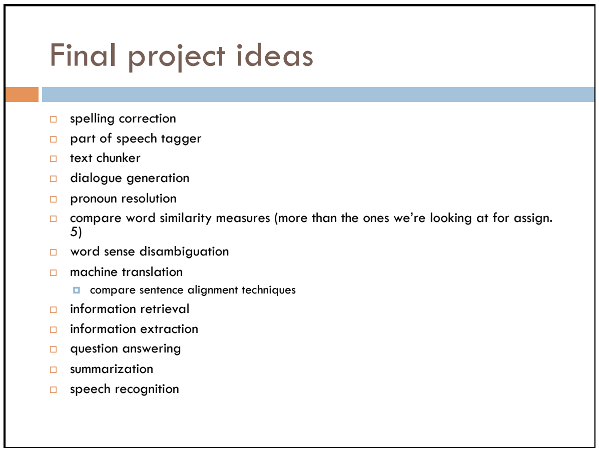
MACHINE LEARNING BASICS

David Kauchak
CS457 Fall 2011



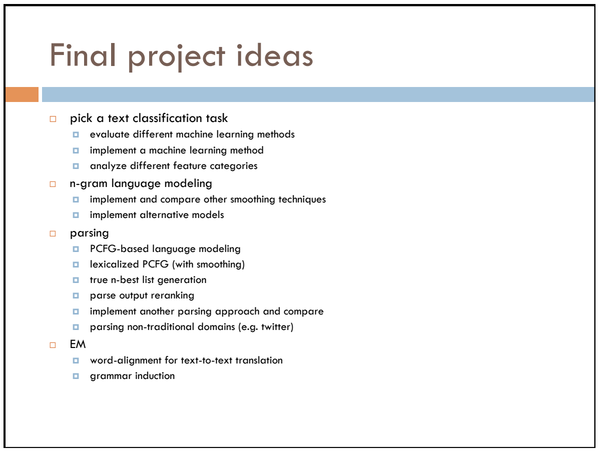
Admin

- Assignment 4
  - How'd it go?
  - How much time?
- Assignment 5
  - Last assignment!
  - "Written" part due Friday
  - Rest, due next Friday
- Read article for discussion on Thursday



Final project ideas

- spelling correction
- part of speech tagger
- text chunker
- dialogue generation
- pronoun resolution
- compare word similarity measures (more than the ones we're looking at for assign. 5)
- word sense disambiguation
- machine translation
  - compare sentence alignment techniques
- information retrieval
- information extraction
- question answering
- summarization
- speech recognition



Final project ideas

- pick a text classification task
  - evaluate different machine learning methods
  - implement a machine learning method
  - analyze different feature categories
- n-gram language modeling
  - implement and compare other smoothing techniques
  - implement alternative models
- parsing
  - PCFG-based language modeling
  - lexicalized PCFG (with smoothing)
  - true n-best list generation
  - parse output reranking
  - implement another parsing approach and compare
  - parsing non-traditional domains (e.g. twitter)
- EM
  - word-alignment for text-to-text translation
  - grammar induction

Word similarity

- Four general categories
  - Character-based
    - turned vs. truned
    - cognates (night, nacht, nicht, natt, nat, noc, noch)
  - Semantic web-based (e.g. WordNet)
  - Dictionary-based
  - Distributional similarity-based
    - similar words occur in similar contexts

Corpus-based approaches

Word

ANY blurb

aardvark

beagle

dog

Corpus-based

The ***Beagle*** is a breed of small to medium-sized dog. A member of the Hound Group, it is similar in appearance to the Foxhound but smaller, with shorter leg

Beagles are intelligent, and are popular as pets because of their size, even temper, and lack of inherited health problems.

Dogs of similar size and purpose to the modern ***Beagle*** can be traced in Ancient Greece[2] back to around the 5th century BC.

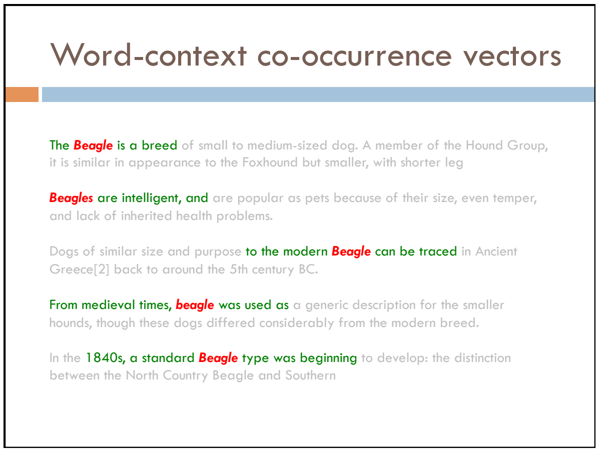
From medieval times, ***beagle*** was used as a generic description for the smaller hounds, though these dogs differed considerably from the modern breed.

In the 1840s, a standard ***Beagle*** type was beginning to develop: the distinction between the North Country Beagle and Southern

Corpus-based: feature extraction

The ***Beagle*** is a breed of small to medium-sized dog. A member of the Hound Group, it is similar in appearance to the Foxhound but smaller, with shorter leg

- We'd like to utilize or vector-based approach
- How could we we create a vector from these occurrences?
  - collect word counts from all documents with the word in it
  - collect word counts from all sentences with the word in it
  - collect all word counts from all words within ***X*** words of the word
  - collect all words counts from words in specific relationship: subject-object, etc.



Word-context co-occurrence vectors

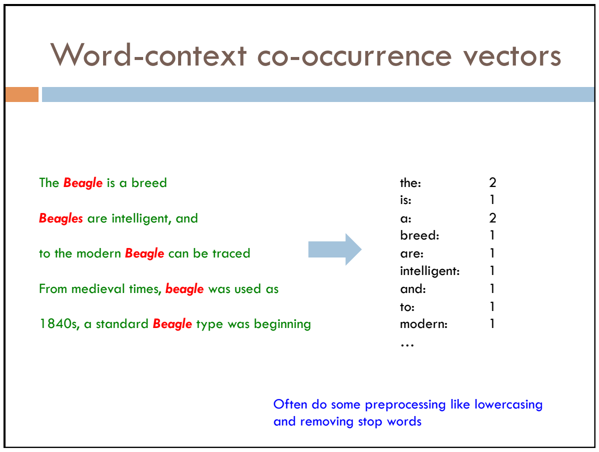
The ***Beagle*** is a breed of small to medium-sized dog. A member of the Hound Group, it is similar in appearance to the Foxhound but smaller, with shorter leg

Beagles are intelligent, and are popular as pets because of their size, even temper, and lack of inherited health problems.

Dogs of similar size and purpose to the modern ***Beagle*** can be traced in Ancient Greece[2] back to around the 5th century BC.

From medieval times, ***beagle*** was used as a generic description for the smaller hounds, though these dogs differed considerably from the modern breed.

In the 1840s, a standard ***Beagle*** type was beginning to develop: the distinction between the North Country Beagle and Southern



Word-context co-occurrence vectors

The ***Beagle*** is a breed

Beagles are intelligent, and

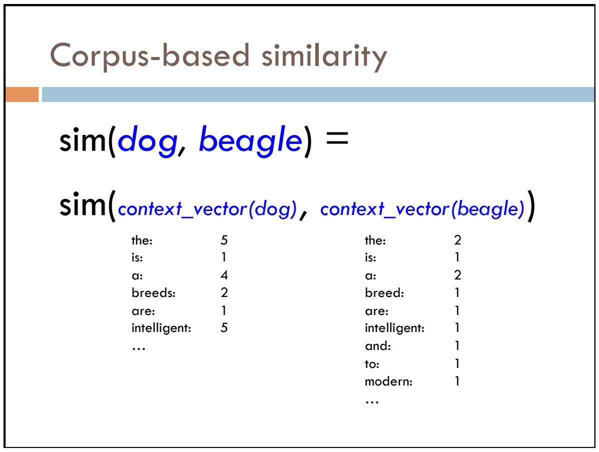
to the modern ***Beagle*** can be traced

From medieval times, ***beagle*** was used as

1840s, a standard ***Beagle*** type was beginning

the:	2
is:	1
a:	2
breed:	1
are:	1
intelligent:	1
and:	1
to:	1
modern:	1
…	

Often do some preprocessing like lowercasing and removing stop words



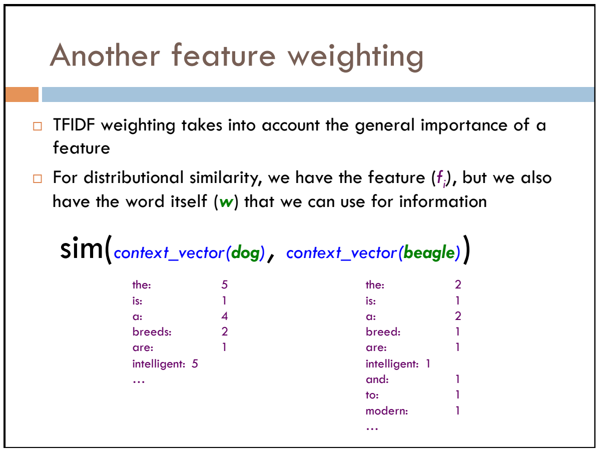
Corpus-based similarity

sim(*dog*, *beagle*) =

sim(*context_vector(dog)*, *context_vector(beagle)*)

the:	5
is:	1
a:	4
breeds:	2
are:	1
intelligent:	5
…	

the:	2
is:	1
a:	2
breed:	1
are:	1
intelligent:	1
and:	1
to:	1
modern:	1
…	



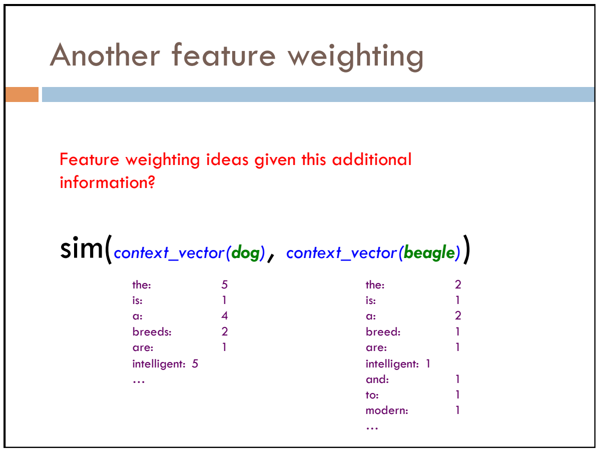
Another feature weighting

- TFIDF weighting takes into account the general importance of a feature
- For distributional similarity, we have the feature (f_i), but we also have the word itself (***w***) that we can use for information

sim(*context_vector(**dog**)*, *context_vector(**beagle**)*)

the:	5
is:	1
a:	4
breeds:	2
are:	1
intelligent:	5
…	

the:	2
is:	1
a:	2
breed:	1
are:	1
intelligent:	1
and:	1
to:	1
modern:	1
…	

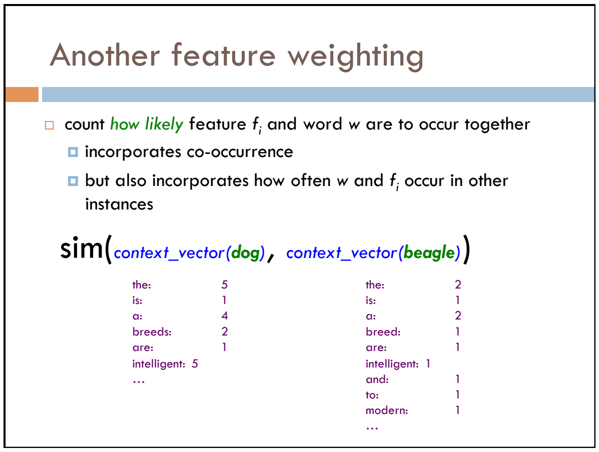


Another feature weighting

Feature weighting ideas given this additional information?

sim(*context_vector*(***dog***), *context_vector*(***beagle***))

the:	5	the:	2
is:	1	is:	1
a:	4	a:	2
breeds:	2	breed:	1
are:	1	are:	1
intelligent:	5	intelligent:	1
…		and:	1
		to:	1
		modern:	1
		…	

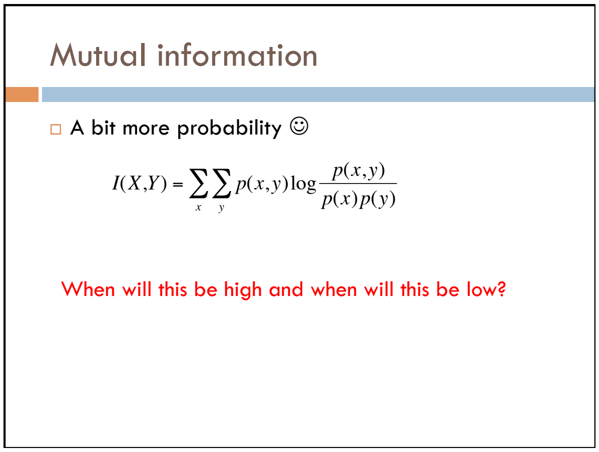


Another feature weighting

- count *how likely* feature f_i and word w are to occur together
  - incorporates co-occurrence
  - but also incorporates how often w and f_i occur in other instances

sim(*context_vector*(***dog***), *context_vector*(***beagle***))

the:	5	the:	2
is:	1	is:	1
a:	4	a:	2
breeds:	2	breed:	1
are:	1	are:	1
intelligent:	5	intelligent:	1
…		and:	1
		to:	1
		modern:	1
		…	

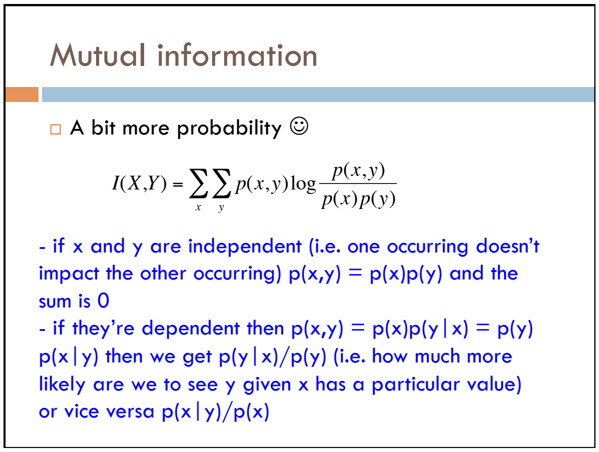


Mutual information

- A bit more probability ☺

$$I(X,Y) = \sum_x \sum_y p(x,y)\log\frac{p(x,y)}{p(x)p(y)}$$

When will this be high and when will this be low?



Mutual information

- A bit more probability ☺

$$I(X,Y) = \sum_x \sum_y p(x,y)\log\frac{p(x,y)}{p(x)p(y)}$$

- if x and y are independent (i.e. one occurring doesn't impact the other occurring) p(x,y) = p(x)p(y) and the sum is 0

- if they're dependent then p(x,y) = p(x)p(y|x) = p(y) p(x|y) then we get p(y|x)/p(y) (i.e. how much more likely are we to see y given x has a particular value) or vice versa p(x|y)/p(x)

Point-wise mutual information

Mutual information

$$I(X,Y) = \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

How related are two variables (i.e. over all possible values/events)

Point-wise mutual information

$$PMI(x,y) = \log \frac{p(x,y)}{p(x)p(y)}$$

How related are two events/values

PMI weighting

- Mutual information is often used for feature selection in many problem areas
- PMI weighting weights co-occurrences based on their correlation (i.e. high PMI)

context_vector(beagle)

the:	2
is:	1
a:	2
breed:	1
are:	1
intelligent:	1
and:	1
to:	1
modern:	1
…	

$\log \frac{p(beagle,the)}{p(beagle)p(the)}$ this would likely be lower

$\log \frac{p(beagle,breed)}{p(beagle)p(breed)}$ this would likely be higher

The mind-reading game

How good are you at guessing random numbers?

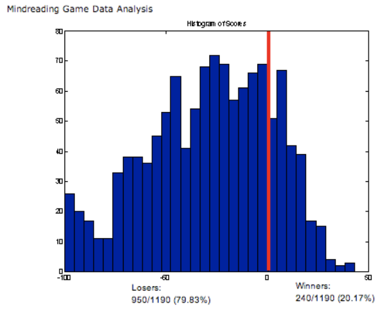
```
Repeat 100 times:
  Computer guesses whether you'll type 0/1
  You type 0 or 1
```

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"!

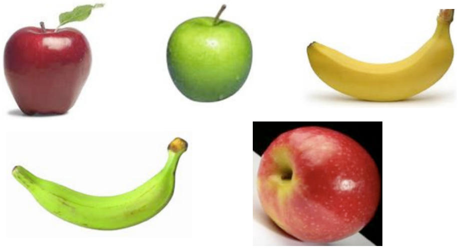
Why machine learning?

- 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

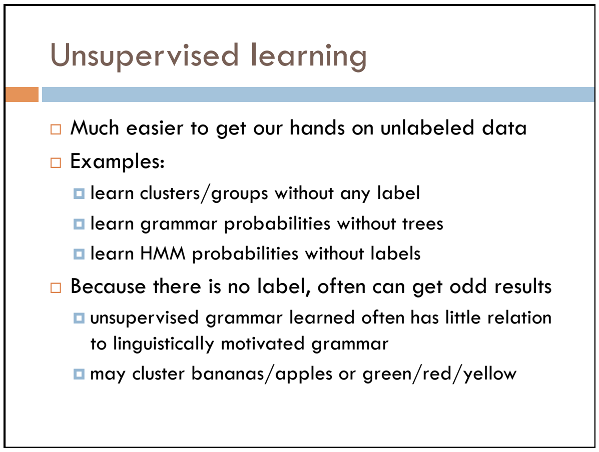
Machine learning problems

- 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. discriminative
    - parametric vs. non-parametric
    - SVM, NB, decision tree, k-means
  - What are we trying to predict:
    - classification vs. regression

Unsupervised learning

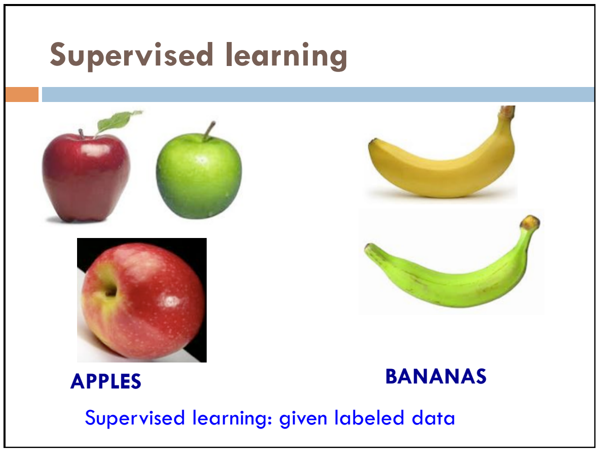


Unupervised learning: given data, but no labels



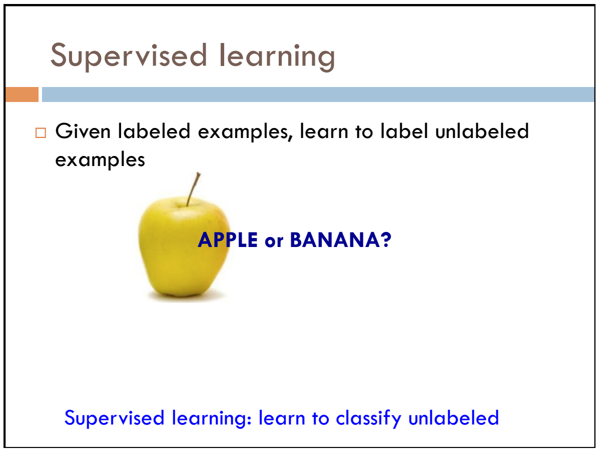
Unsupervised learning

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



Supervised learning

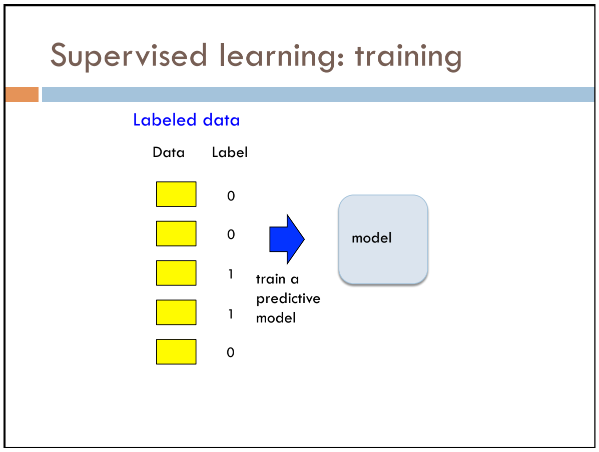
Supervised learning: given labeled data



Supervised learning

- Given labeled examples, learn to label unlabeled examples

Supervised learning: learn to classify unlabeled

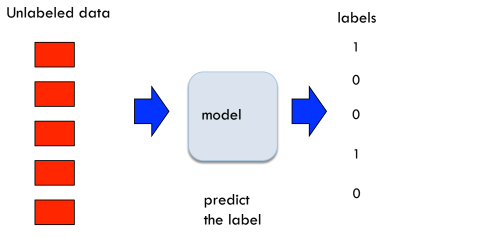


Supervised learning: training

Labeled data

Data	Label
🟨	0
🟨	0
🟨	1
🟨	1
🟨	0

Supervised learning: testing/classifying



Unlabeled data

labels

1

0

0

1

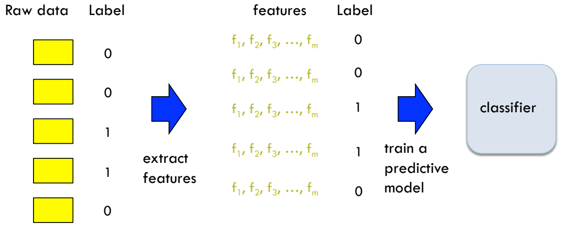
0

model

predict
the label

Feature based learning

Training or learning phase



Raw data

Label

0

0

1

1

0

extract
features

features

Label

$f_1, f_2, f_3, \ldots, f_m$ 0

$f_1, f_2, f_3, \ldots, f_m$ 0

$f_1, f_2, f_3, \ldots, f_m$ 1

$f_1, f_2, f_3, \ldots, f_m$ 1

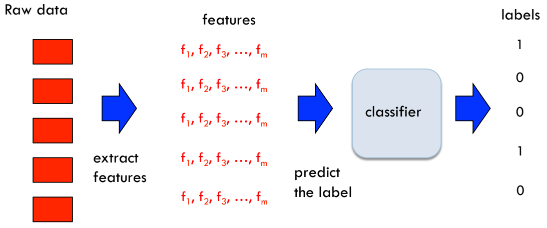
$f_1, f_2, f_3, \ldots, f_m$ 0

train a
predictive
model

classifier

Feature based learning

Testing or classification phase



Raw data

extract
features

features

$f_1, f_2, f_3, \ldots, f_m$

$f_1, f_2, f_3, \ldots, f_m$

$f_1, f_2, f_3, \ldots, f_m$

$f_1, f_2, f_3, \ldots, f_m$

$f_1, f_2, f_3, \ldots, f_m$

classifier

predict
the label

labels

1

0

0

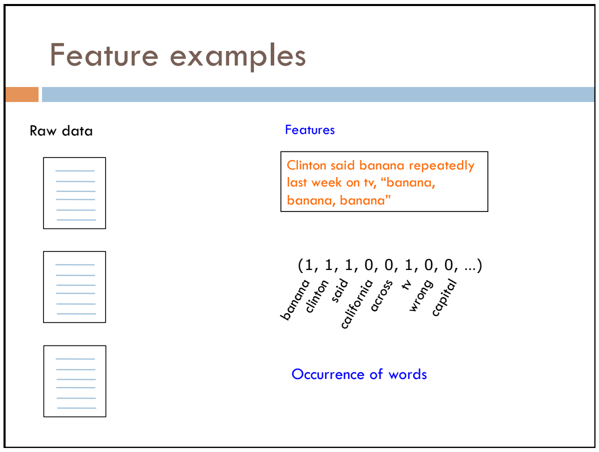
1

0

Feature examples

Raw data

Features?



Feature examples

Raw data

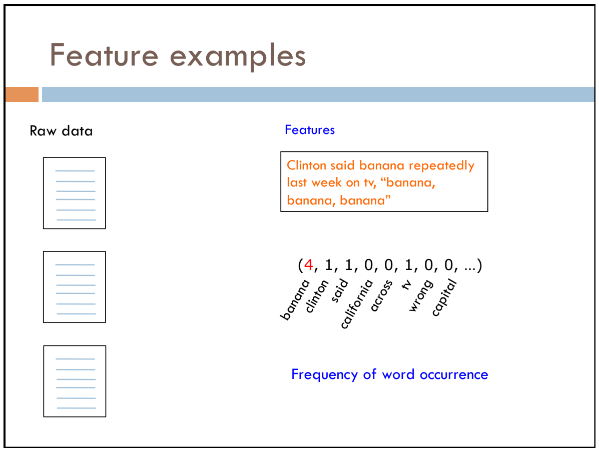
Features

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

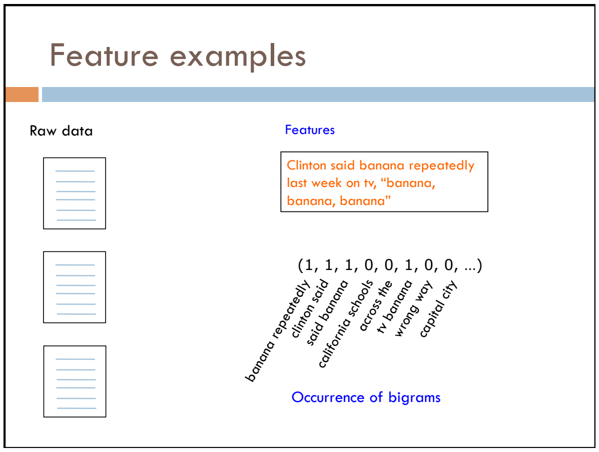
Features

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

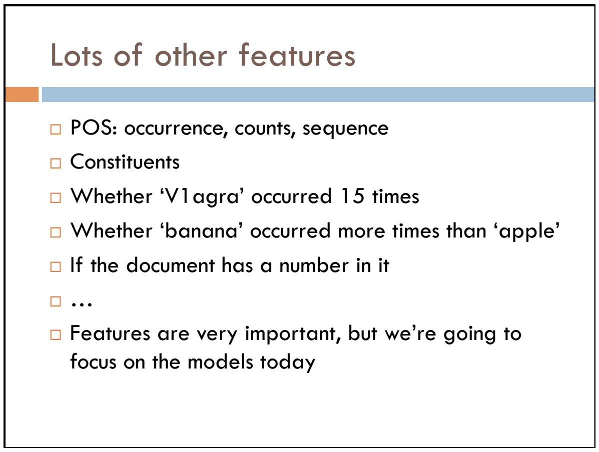
Features

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

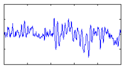


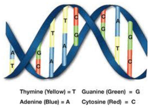
Lots of other features

- 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

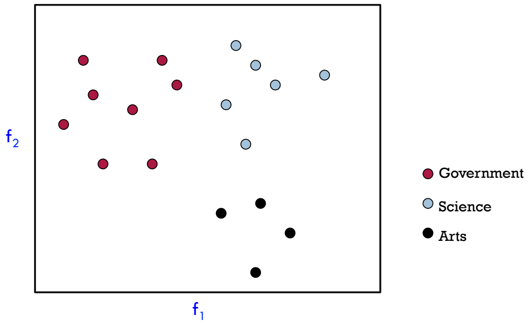
Power of feature-base methods

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

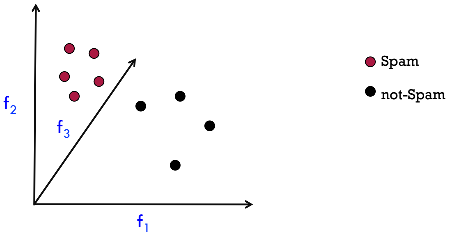




The feature space



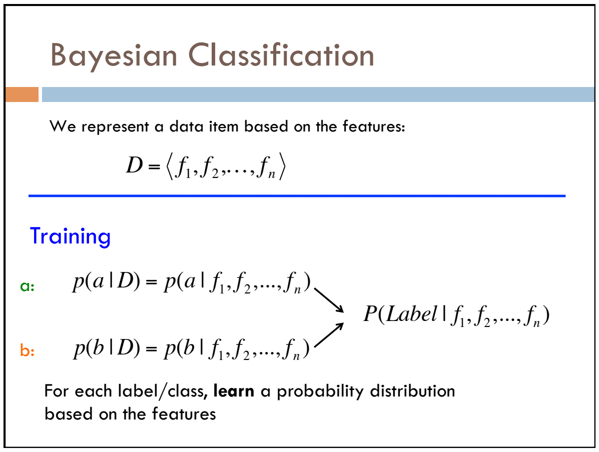
The feature space



Feature space

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

How big will m be for us?



Bayesian Classification

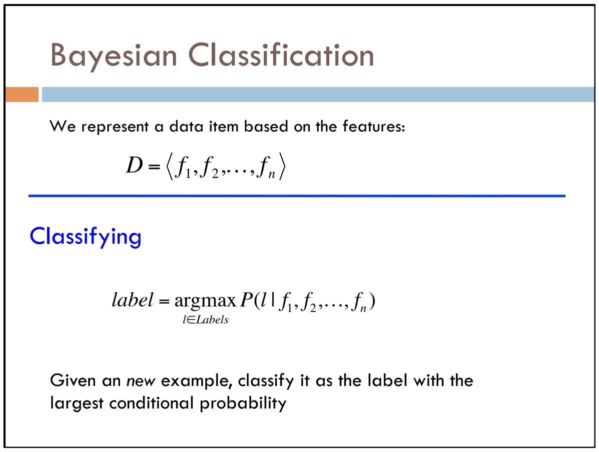
We represent a data item based on the features:

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

Training

$$\begin{tikzcd} \text{a:}\ p(a|D) = p(a|f_1,f_2,...,f_n) \arrow[dr] & \\ & P(Label|f_1,f_2,...,f_n) \\ \text{b:}\ p(b|D) = p(b|f_1,f_2,...,f_n) \arrow[ur] & \end{tikzcd}$$

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



Bayesian Classification

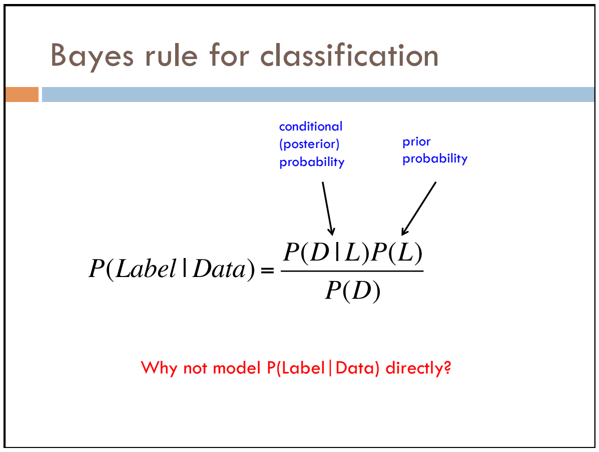
We represent a data item based on the features:

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

Classifying

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

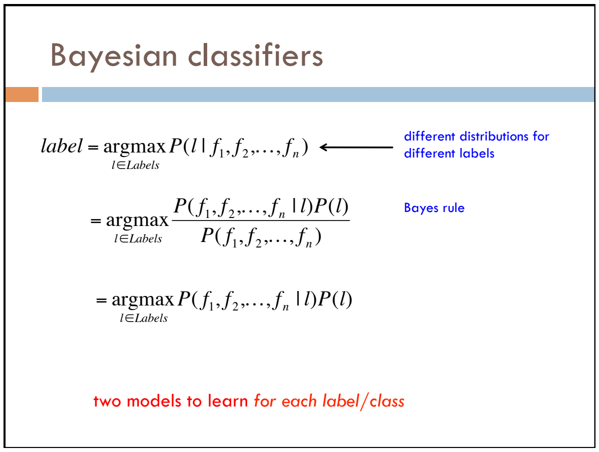
Given an *new* example, classify it as the label with the largest conditional probability



Bayes rule for classification

$$P(Label|Data) = \frac{\overset{\text{conditional (posterior) probability}}{P(D|L)}\ \overset{\text{prior probability}}{P(L)}}{P(D)}$$

Why not model P(Label|Data) directly?



Bayesian classifiers

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

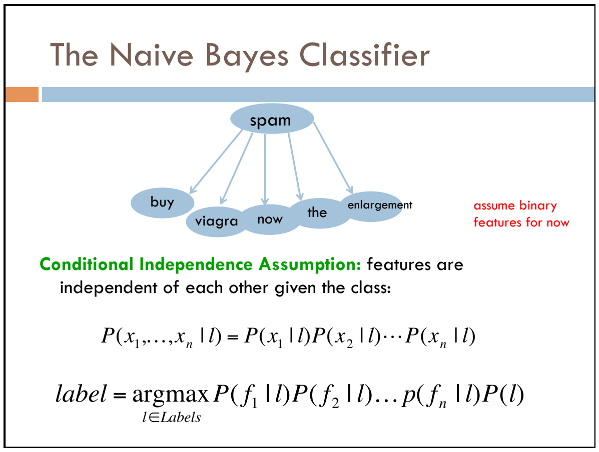
different distributions for different labels

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

Bayes rule

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

two models to learn *for each label/class*



The Naive Bayes Classifier

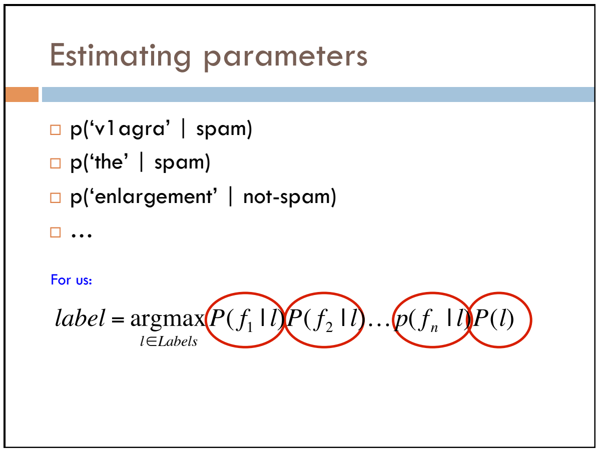
spam
buy
viagra
now
the
enlargement

assume binary features for now

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

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

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

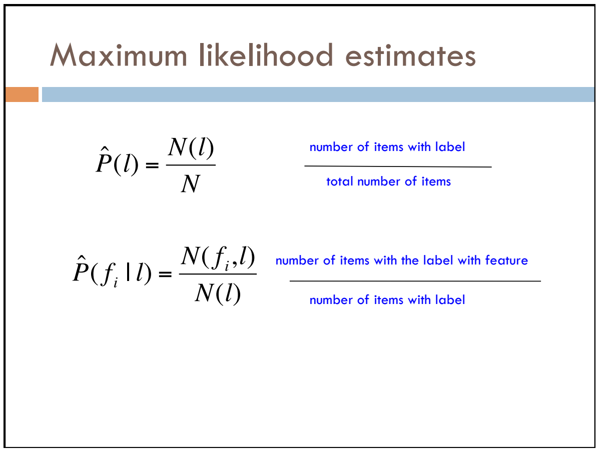


Estimating parameters

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

For us:

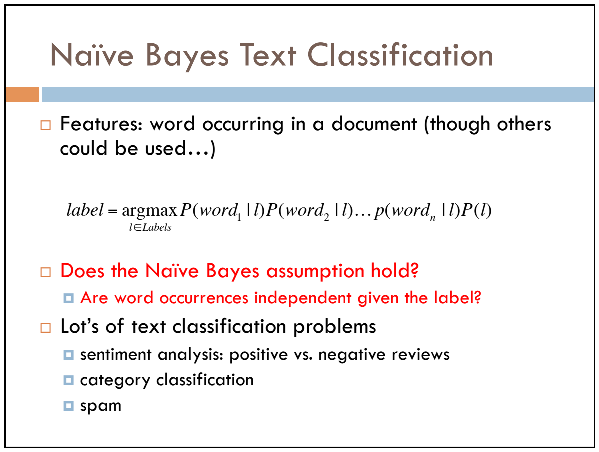
$$label = \underset{l \in Labels}{\operatorname{argmax}} P(f_1 \mid l)P(f_2 \mid l)\ldots p(f_n \mid l)P(l)$$



Maximum likelihood estimates

$$\hat{P}(l) = \frac{N(l)}{N} \qquad \frac{\text{number of items with label}}{\text{total number of items}}$$

$$\hat{P}(f_i \mid l) = \frac{N(f_i,l)}{N(l)} \qquad \frac{\text{number of items with the label with feature}}{\text{number of items with label}}$$



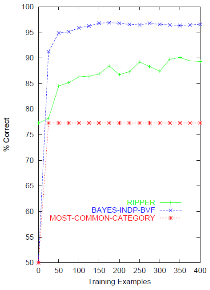
Naïve Bayes Text Classification

- 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



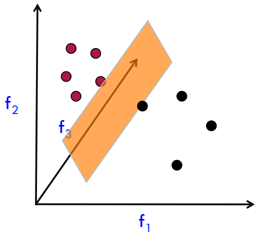
http://www.cnbc.cmu.edu/~jp/research/email.paper.pdf

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

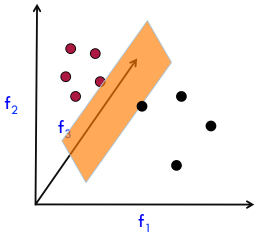


Linear models

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

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

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

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

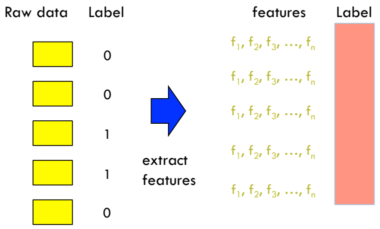
$$= \underset{l \in Labels}{\operatorname{argmax}} \log(P(f_1 | l)) + \log(P(f_2 | l)) + \ldots + \log(p(f_n | l)) + \log(P(l))$$

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

$f_1 w_1$ $f_2 w_2$ w_0

only one of f_1 and f_2 will every be 1

Regression vs. classification


Raw data
Label
0
0
1
1
0
extract
features
features
Label
$f_1, f_2, f_3, \ldots, f_n$
$f_1, f_2, f_3, \ldots, f_n$
$f_1, f_2, f_3, \ldots, f_n$
$f_1, f_2, f_3, \ldots, f_n$
$f_1, f_2, f_3, \ldots, f_n$

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

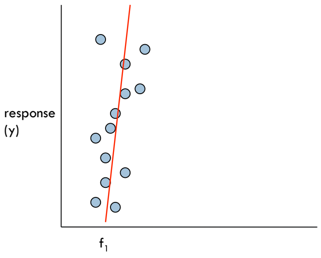
Examples

- 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

- A model
  - Often we have an idea of what the data might look like
  - … or we don't, but we assume the data looks like something we know how to handle
- Learning then involves finding the best parameters for the model based on the data
- Regression models (describe how the features combine to get the result/label)
  - linear
  - logistic
  - polynomial
  - …

Linear regression

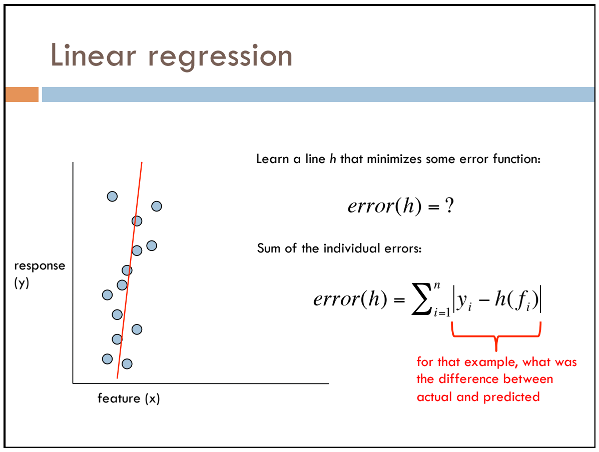


How can we find this line?

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

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



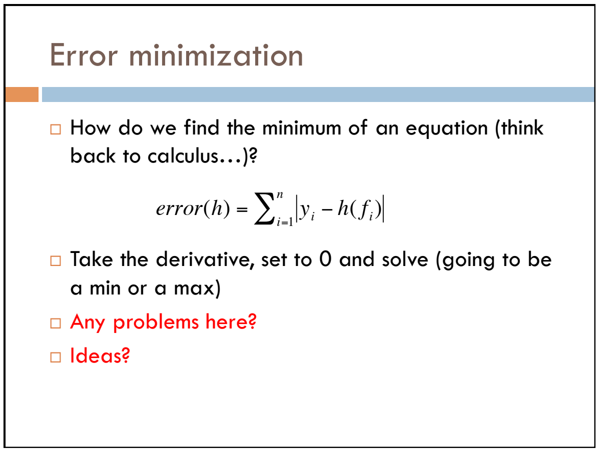
Linear regression

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

$$error(h) = ?$$

Sum of the individual errors:

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

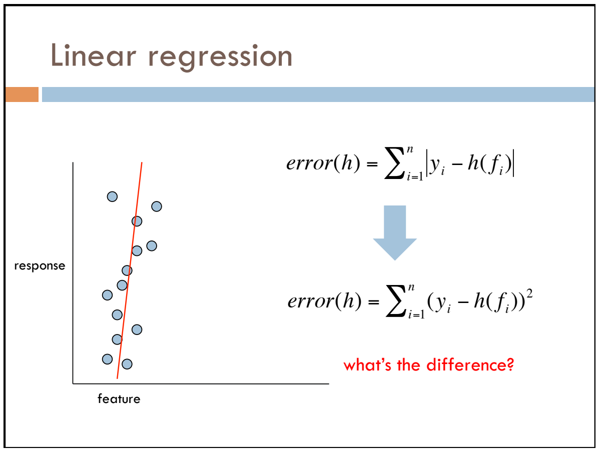


Error minimization

- 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?



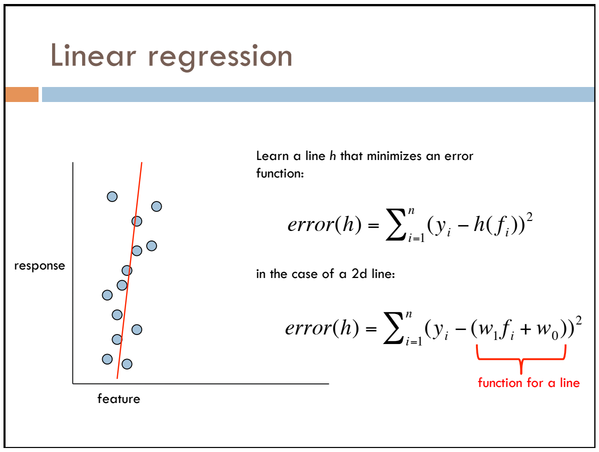
Linear regression

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

$$\Downarrow$$

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

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

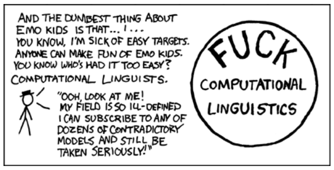
$$h(\bar{f}) = w_0 + w_1 f_1 + w_2 f_2 + \ldots + w_m f_m$$

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

Still can solve this exactly!

Probabilistic classification

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


AND THE DUMBEST THING ABOUT EMO KIDS IS THAT... I... YOU KNOW, I'M SICK OF EASY TARGETS. ANYONE CAN MAKE FUN OF EMO KIDS. YOU KNOW WHO'S HAD IT TOO EASY? COMPUTATIONAL LINGUISTS.
"OOH, LOOK AT ME! MY FIELD IS SO ILL-DEFINED I CAN SUBSCRIBE TO ANY OF DOZENS OF CONTRADICTORY MODELS AND STILL BE TAKEN SERIOUSLY!"
FUCK
COMPUTATIONAL
LINGUISTICS

- 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

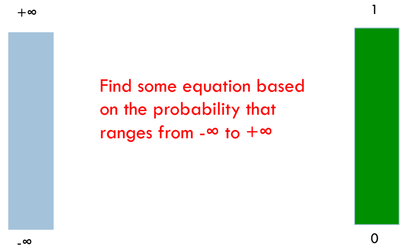
The challenge

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

Linear regression

$p(1 \mid x_1, x_2, \ldots, x_m)$

probability



Find some equation based on the probability that ranges from -∞ to +∞

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?

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

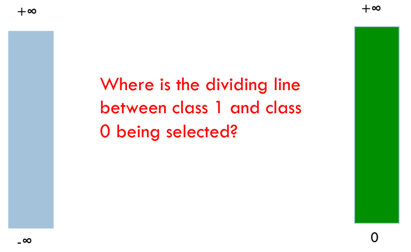
Odds ratio

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

Linear regression

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

odds ratio



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

Odds ratio

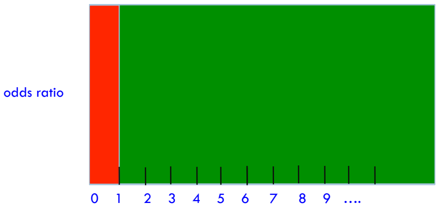
$P(1 | x_1,x_2,...,x_m) > P(0 | x_1,x_2,...,x_m)$

$$\frac{P(1 | x_1,x_2,...,x_m)}{1 - P(1 | x_1,x_2,...,x_m)}$$

$P(1 | x_1,x_2,...,x_m) > 1 - P(1 | x_1,x_2,...,x_m)$

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

Does this suggest another transformation?



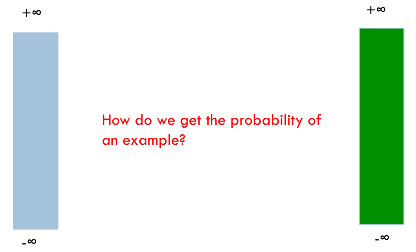
Log odds (logit function)

$w_0 + x_1 w_1 + w_2 x_2 + ... + w_m x_m$

$$\log \frac{P(1 | x_1,x_2,...,x_m)}{1 - P(1 | x_1,x_2,...,x_m)}$$

Linear regression

odds ratio



How do we get the probability of an example?

Log odds (logit function)

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

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

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

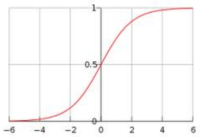
$$\ldots$$

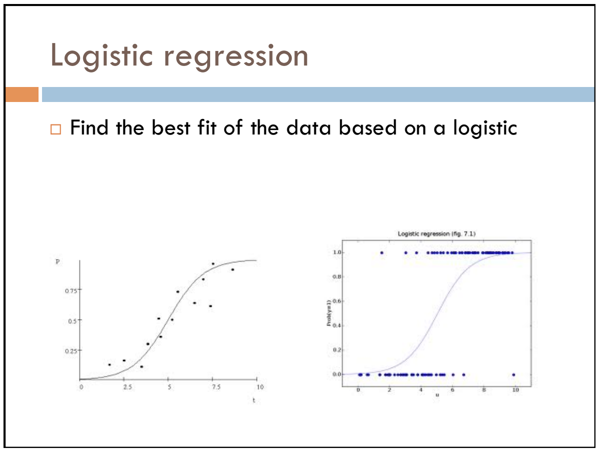
$$P(1 | x_1,x_2,...,x_m) = \frac{1}{1 + e^{-(w_0 + w_1 x_2 + w_2 x_2 + ... + w_m x_m)}}$$

anyone recognize this?

Logistic function

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





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?

$$\log\frac{P(1 \mid x_1,x_2,\ldots,x_m)}{1-P(1 \mid x_1,x_2,\ldots,x_m)} = w_0 + w_1x_2 + w_2x_2 + \ldots + w_mx_m$$

- □ 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)