Text Mining

Week 3.

Classification, part 1

Is it spam?

http://jsanchezs.com/auth.berkley.edu/cas/loginc357.html

From: "HR@berkeley.edu" <HR@berkeley.edu>
Subject: Message from human resources
Date: April 13, 2017 at 9:29:54 PM PDT
To: XXXXX@berkeley.edu

Dear XXXXX@berkeley.edu

An information document has been sent to you by the Human Resources Department.

Click here to Login to view the document. Thank you!

Berkeley University Of California HR Department

Is it spam?
Why do you think so?

Authorship attribution

- 1787-8: anonymous essays try to convince New York to ratify U.S. Constitution: Jay, Madison, Hamilton
- Authorship of 12 of the letters in dispute
- 1963: solved using Bayesian methods (paper listed for week 1).

Male of Female author?

- Number of pronouns, determiners, noun phrases

Sentiment analysis: positive or negative

- Positive of negative?
  - Unbelievably disappointing
  - Full or zany characters and richly applied satire, and some great plot twists
  - It is the greatest screwball comedy ever filmed
  - It was pathetic. The worst part about it was the boxing scenes.

What is the subject of the article.

Dan Jurafsky

What is the subject of this article?

MEDLINE Article

?

MeSH Subject Category Hierarchy

- Antogonists and Inhibitors
- Blood Supply
- Chemistry
- Drug Therapy
- Embryology
- Epidemiology
- …

Text Categorization Applications

- Web pages
  - Recommending
  - Yahoo-like classification
- Newsgroup/Blog Messages
  - Recommending
  - spam filtering (we do not see 99% of spam in our mail box)
  - Sentiment analysis for marketing
- News articles
  - Personalized newspaper
- Authorship attribution
- Age / gender identification
- Language identification
- Sentiment analysis

Formal definition of Text Classification

- *Input*:
  - A document *d*
  - A fixed set of classes $C = \{c1, c2, c3, \ldots cj\}$

- *Output*: a predicted class $c \in C$

Take a document and assign a class to this document

Classification Methods: hand-coded rules

- Rules based on combinations of words or other features
  - Spam: black-list-address OR (“dollars” and “have been selected”)
- Accuracy can be high
  - If rules are carefully created and/or refined by expert
- **But** building and maintaing these rules is expensive

→ Machine learning methods enter the stage

Supervised Machine Learning

- *Input*:
  - A document *d*
  - A fixed set of classes $C = \{c1, c2, c3, \ldots cj\}$
  - NEW: A training set of *m* hand-labeled documents *(d1,c1)….(dm,cn)*

- *Output*: a learned classifier $\gamma: d \rightarrow \in C$

Classification

- Document class *t*, target class
- Document representation is *x*, the feature vector
- Goal: learn

$$f(x) = t$$

Classification methods

- Now that we have a way to express a document *x*, let's do document classification.
- Assume we have some labeled data for training. (i.e. known paired *x* and *t*)
  - Decision trees
  - Naïve Bayes
  - Logistic regression
  - Neural nets

Classification methods: Rule-based

- There are “IDE” type development environments for writing very complex rules efficiently.
- Often: Boolean combinations (as in Google Alerts)
- Accuracy is very high if a rule has been carefully refined over time by a subject expert.
- Building and maintaining rule-based classification systems is expensive. (does it remind you anything??)

Classification methods: Statistical/Probabilistic

- As per our definition of the classification problem – text classification as a learning problem
- Supervised learning of a the classification function and its application to classifying new documents
- We will look at a couple of methods for doing this: Naive Bayes, Logistic Regression, SVM, Decision Trees
- No free lunch: requires hand-classified training data
- But this manual classification can be done by non-experts

Generative vs. Discriminative Models

- Generative models: generate the observed data from hidden stuff
  - Goal, given observation x, compute probability of label y, p(y|x)
  - Naïve Bayes (later) uses Bayes rule to reverse conditioning

- What if we care about p(y|x)? We need a more general framework …
- That framework is called logistic regression
  - Logistic: A special mathematical function it uses
  - Regression: Combines a weight vector with observations to create an answer
  - More general cookbook for building conditional probability distributions
- Logistic regression is a discriminative model

Decision Trees

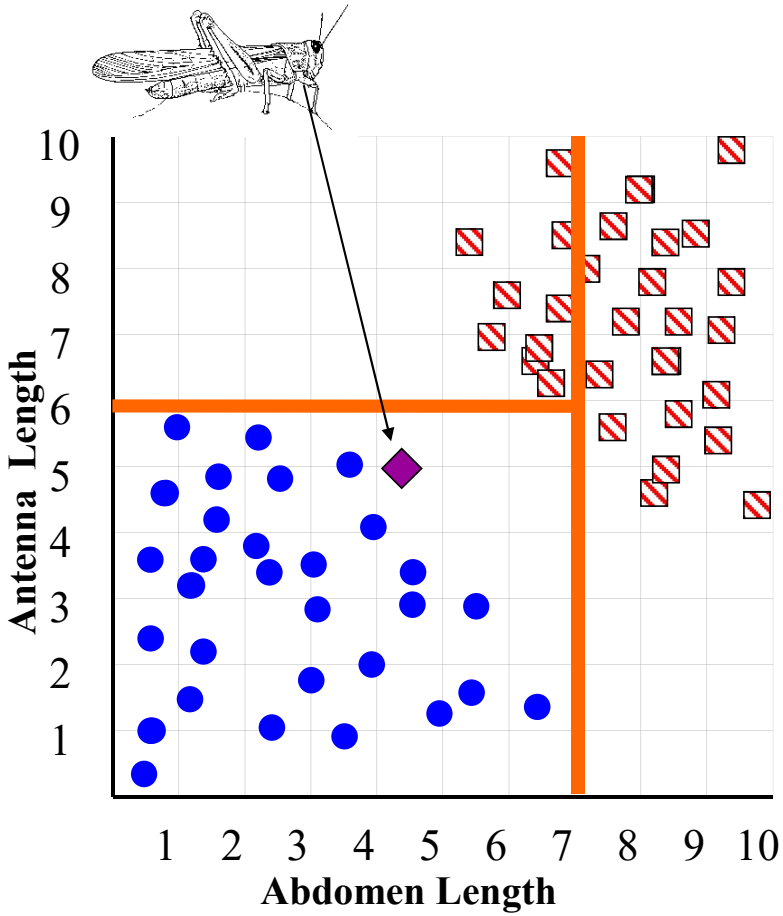
Decision Tree

- Decision trees are powerful and popular tools for classification and prediction.
- Decision trees represent *rules*, which can be understood by humans and used in knowledge system such as database.

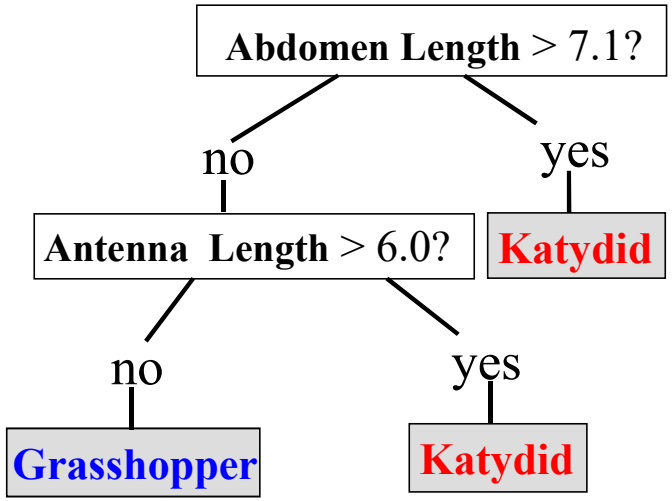
Key Requirements

- **Attribute-value description:** object or case must be expressible in terms of a fixed collection of properties or attributes (e.g., hot, mild, cold).
- **Predefined classes (target values):** the target function has **discrete output values** (bollean or multiclass)

Decision Tree Classifier

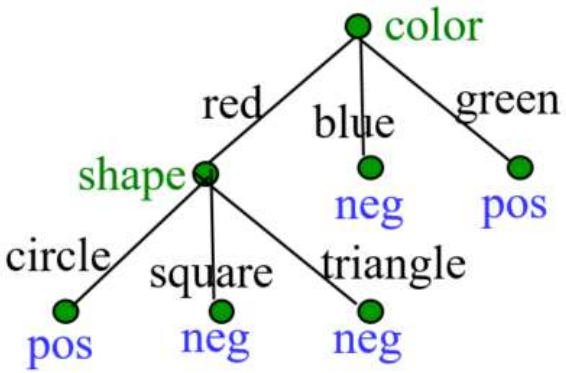


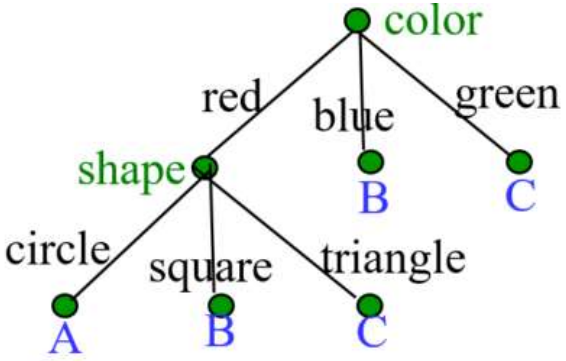
Ross Quinlan



Decision Trees

- Tree-based classifiers represented as feature-vectors. Nodes test features, there is one branch for each value of the feature, and leaves specify the category.





- Can represent arbitrary conjunction and disjunction. Can represent any classification function over discrete feature vectors.
- Can be rewritten as a set of rules, i.e. disjunctive normal form (DNF).
  - red $\wedge$ circle $\rightarrow$ pos
  - red $\wedge$ circle $\rightarrow$ A
    blue $\rightarrow$ B; red $\wedge$ square $\rightarrow$ B
    green $\rightarrow$ C; red $\wedge$ triangle $\rightarrow$ C

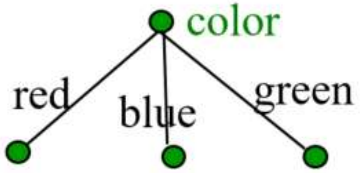
Properties of Decision Tree Learning

- Continuous (real-valued) features can be handled by allowing nodes to split a real valued feature into two ranges based on a threshold (e.g. length < 3 and length ≥3)
- Classification trees have discrete class labels at the leaves, ***regression trees*** allow real-valued outputs at the leaves.
- Methods developed for handling noisy training data (both class and feature noise).
- Methods developed for handling missing feature values.

Top-Down Decision Tree Induction

- Recursively build a tree top-down by divide and conquer.

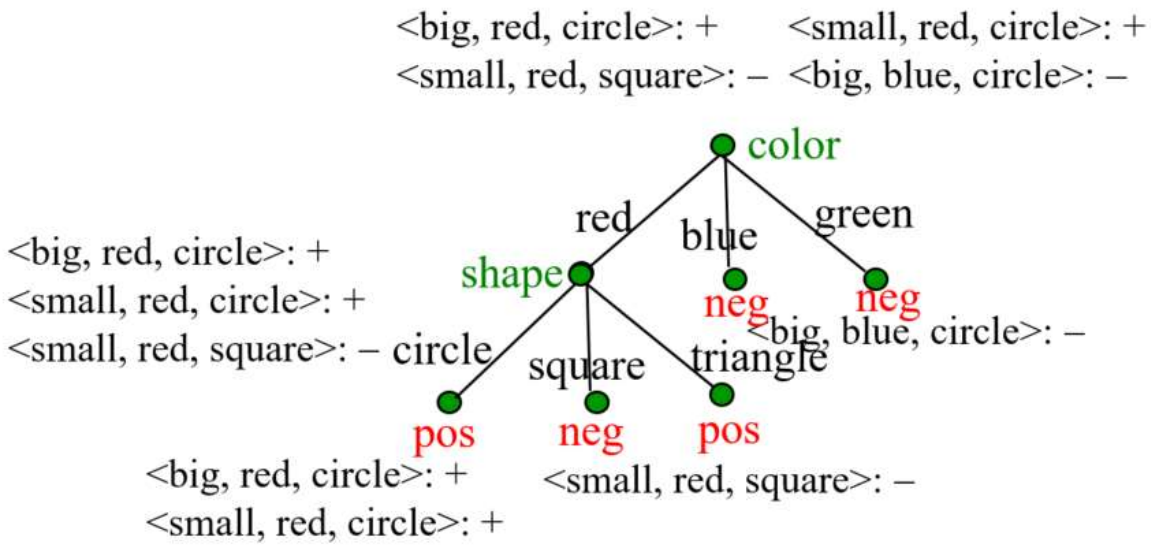
<big, red, circle>: +
<small, red, square>: –
<small, red, circle>: +
<big, blue, circle>: –



<big, red, circle>: +
<small, red, circle>: +
<small, red, square>: –

Top-Down Decision Tree Induction

- Recursively build a tree top-down by divide and conquer.



Decision Tree Induction Pseudocode

DTree(*examples*, *features*) returns a tree
If all *examples* are in one category, return a leaf node with that category label.
Else if the set of *features* is empty, return a leaf node with the category label that
is the most common in examples.
Else pick a feature *F* and create a node *R* for it
For each possible value v_i of *F*:
Let $examples_i$ be the subset of examples that have value v_i for *F*
Add an out-going edge *E* to node *R* labeled with the value v_i.
If $examples_i$ is empty
then attach a leaf node to edge *E* labeled with the category that
is the most common in *examples*.
else call DTree($examples_i$, *features* – {*F*}) and attach the resulting
tree as the subtree under edge *E.*
Return the subtree rooted at *R.*

Decision Tree Induction Pseudocode

- Decision tree induction is a combination of if / else statements
- Given the input set of features it is clear how the decision was made (regarding the class output)
- Easily interpretable

Picking a Good Split Feature

- Goal is to have the resulting tree be as small as possible, per Occam's razor.
- Finding a minimal decision tree (nodes, leaves, or depth) is an NP-hard optimization problem.
- Top-down divide-and-conquer method does a greedy search for a simple tree but does not guarantee to find the smallest.
  - General lesson in ML: "Greed is good."
- Want to pick a feature that creates subsets of examples that are relatively "pure" in a single class so they are "closer" to being leaf nodes.
- There are a variety of heuristics for picking a good test, a popular one is based on information gain that originated with the ID3 system of Quinlan (1979).

Principled Criterion

- Selection of an attribute to test at each node - choosing the most useful attribute for classifying examples.
- ***How***?
- Information gain
  - measures how well a given attribute separates the training examples according to their target classification
  - This measure is used to select among the candidate attributes at each step while growing the tree

Information Gain as a Splitting Criteria

- Select the attribute with the highest information gain (information gain is the expected reduction in entropy).
- Assume there are two classes, *P* and *N*
  - Let the set of examples ***S*** contain *p* elements of class *P* and *n* elements of class *N*
  - The amount of information, needed to decide if an arbitrary example in *S* belongs to *P* or *N* is defined as

$$E(S) = -\frac{p}{p+n}\log_2\left(\frac{p}{p+n}\right) - \frac{n}{p+n}\log_2\left(\frac{n}{p+n}\right)$$

0 log(0) is defined as 0

Entropy

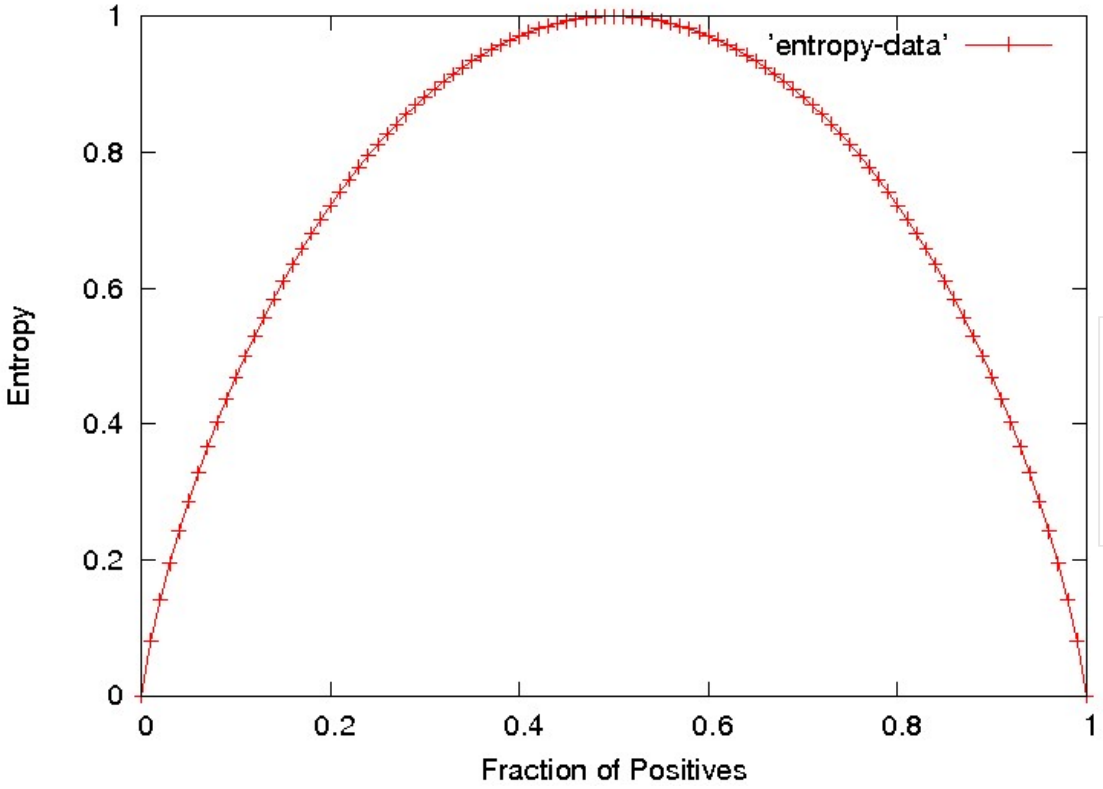
- Entropy (disorder, impurity) of a set of examples, S, relative to a binary classification is:

$$Entropy(S) = -p_1 \log_2(p_1) - p_0 \log_2(p_0)$$

  where p_1 is the fraction of positive examples in S and p_0 is the fraction of negatives.
- If all examples are in one category, entropy is zero (we define $0 \cdot \log(0)=0$)
- If examples are equally mixed ($p_1=p_0=0.5$), entropy is a maximum of 1.
- Entropy can be viewed as the number of bits required on average to encode the class of an example in *S* where data compression (e.g. Huffman coding) is used to give shorter codes to more likely cases.
- For multi-class problems with c categories, entropy generalizes to:

$$Entropy(S) = \sum_{i=1}^{c} -p_i \log_2(p_i)$$

Entropy Plot for Binary Classification



- The entropy is 0 if the outcome is ***certain***.
- The entropy is maximum if we have no knowledge of the system (or any outcome is equally possible).

Entropy of a 2-class problem with regard to the portion of one of the two groups

Information Gain

- The information gain of a feature *F* is the expected reduction in entropy resulting from splitting on this feature.

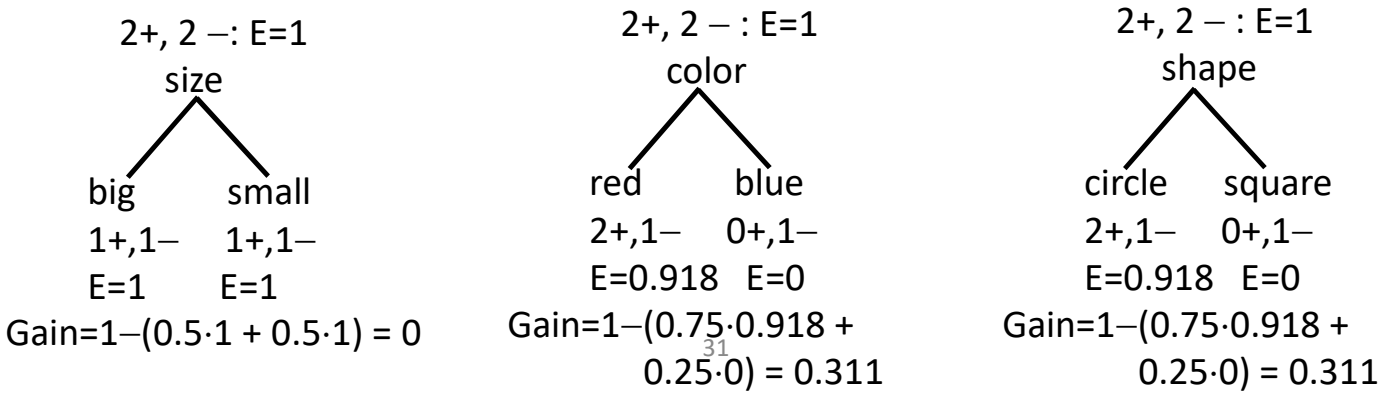
$$Gain(S,F) = Entropy(S) - \sum_{v \in Values(F)} \frac{|S_v|}{|S|} Entropy(S_v)$$

where S_v is the subset of *S* having value *v* for feature *F*.

- Entropy of each resulting subset weighted by its relative size.
- Example:
  - <big, red, circle>: +
  - <small, red, square>: –

<small, red, circle>: +

<big, blue, circle>: –



Bias in Decision-Tree Induction

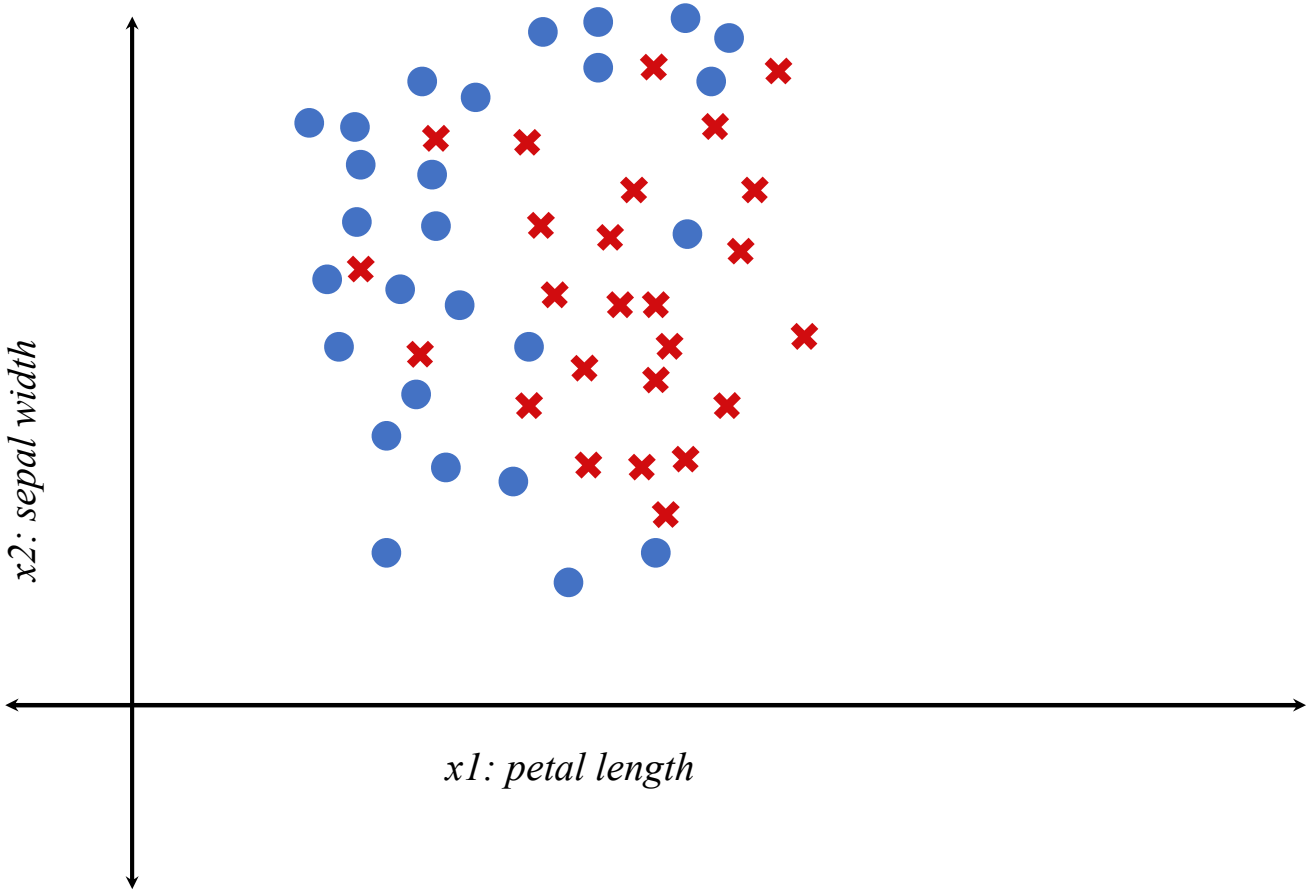
- Information-gain gives a bias for trees with minimal depth.
- Implements a search (preference) bias instead of a language (restriction) bias.

History of Decision-Tree Research

- Hunt and colleagues use exhaustive search decision-tree methods (CLS) to model human concept learning in the 1960's.
- In the late 70's, Quinlan developed ID3 with the information gain heuristic to learn expert systems from examples.
- Simulataneously, Breiman and Friedman and colleagues develop CART (Classification and Regression Trees), similar to ID3.
- In the 1980's a variety of improvements are introduced to handle noise, continuous features, missing features, and improved splitting criteria. Various expert-system development tools results.
- Quinlan's updated decision-tree package (C4.5) released in 1993.

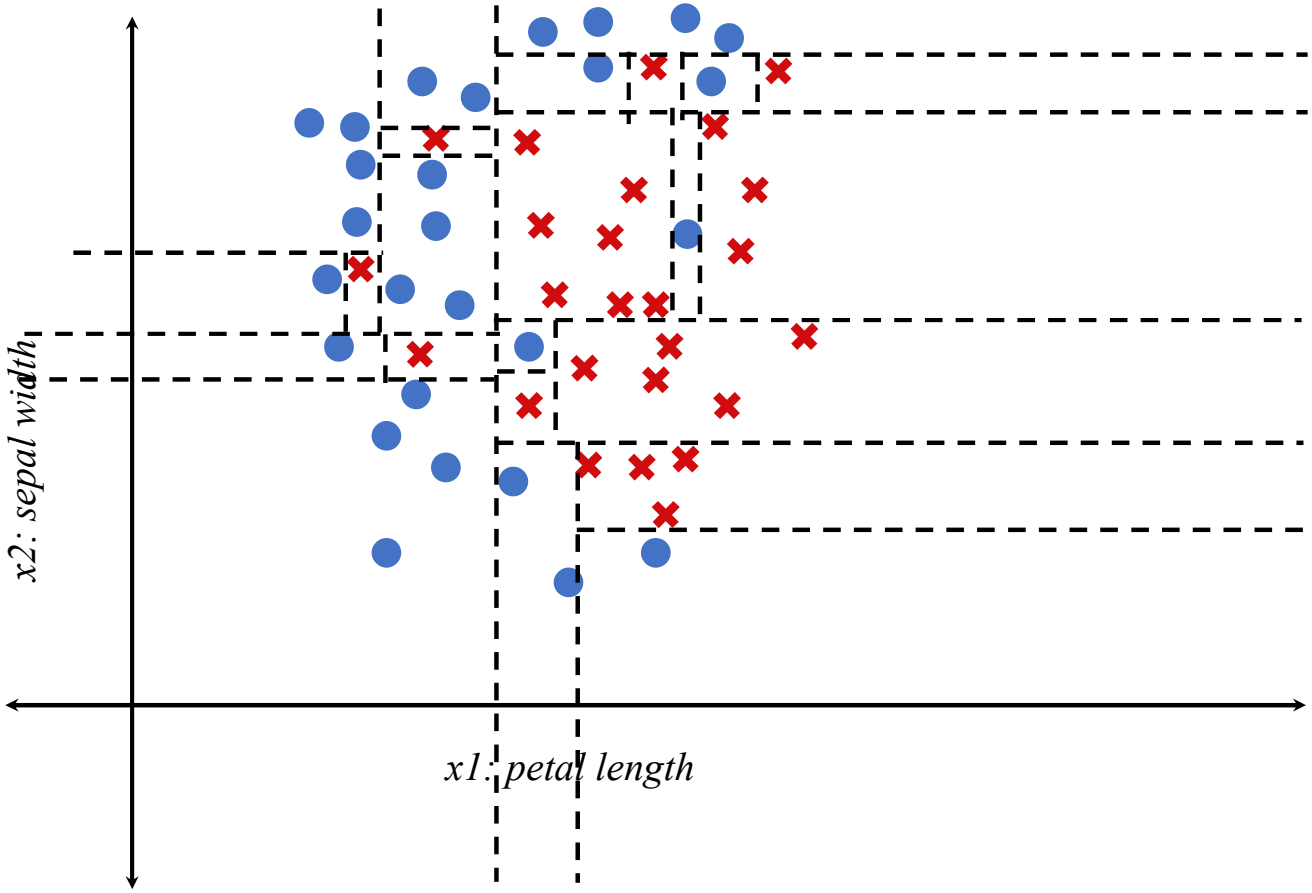
DTs in practice...

- Growing to purity is bad (*overfitting*)



DTs in practice...

- Growing to purity is bad (*overfitting*)

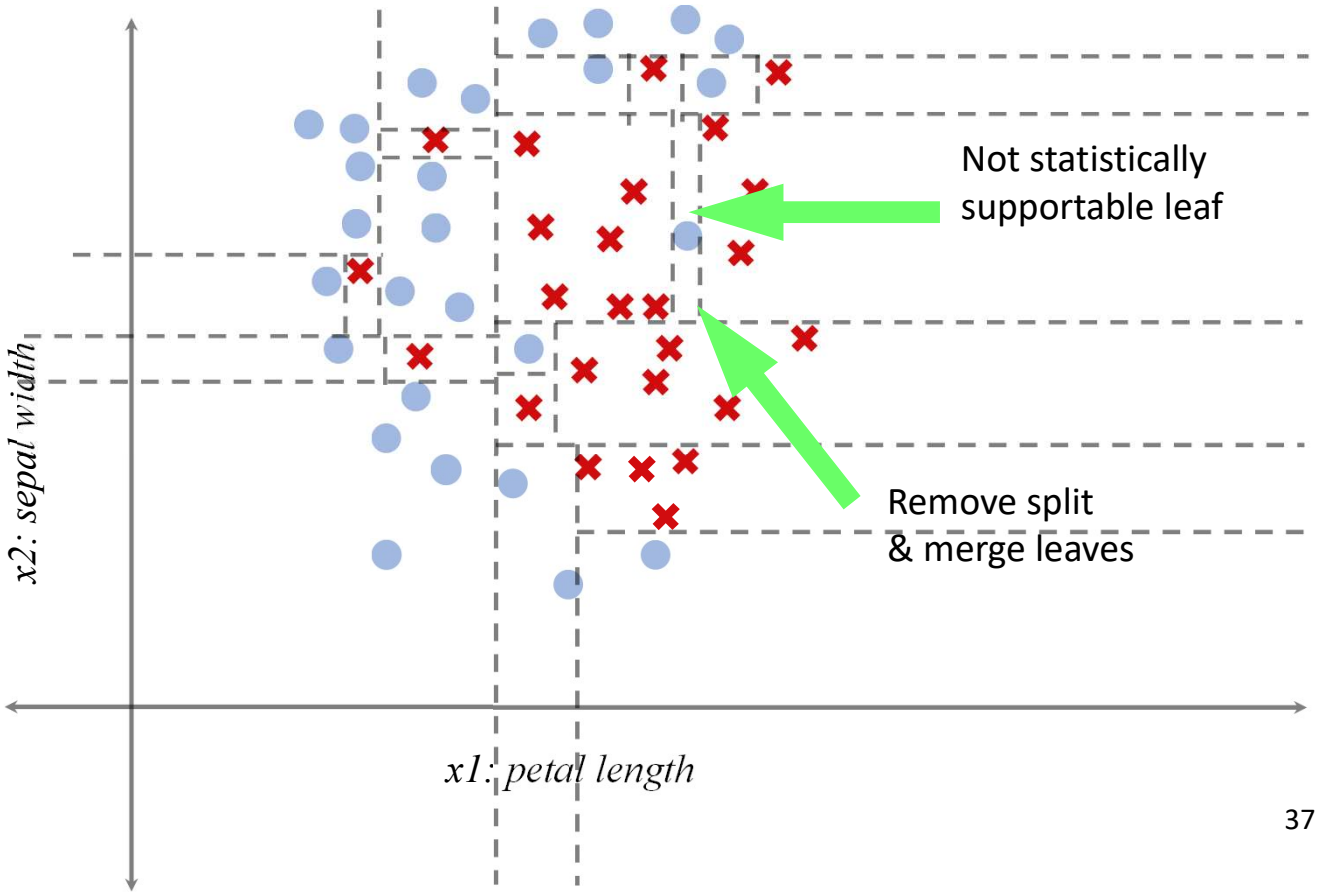


DTs in practice...

- Growing to purity is bad (*overfitting*)
  - Terminate growth early
  - Grow to purity, then *prune* back

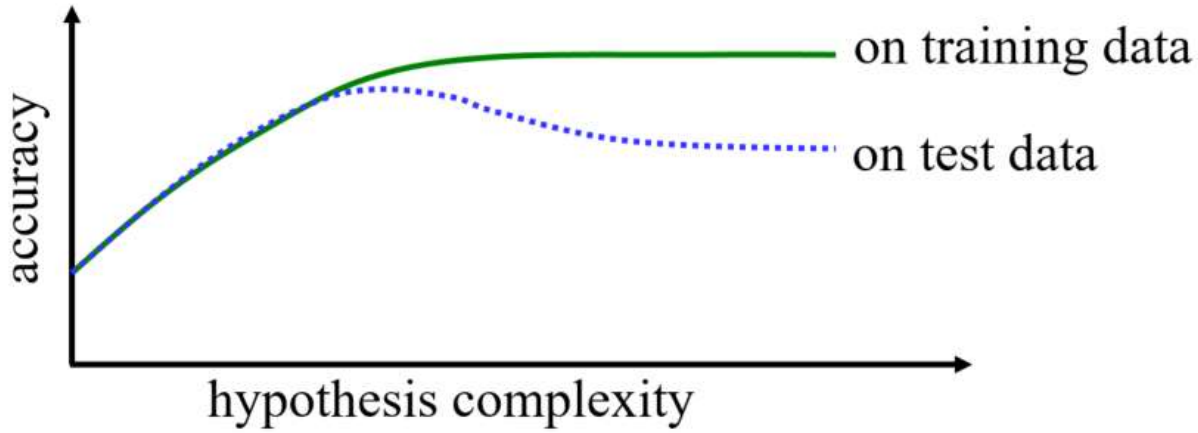
DTs in practice...

- Growing to purity is bad (*overfitting*)



Overfitting

- Learning a tree that classifies the training data perfectly may not lead to the tree with the best generalization to unseen data.
  - There may be noise in the training data that the tree is erroneously fitting.
  - The algorithm may be making poor decisions towards the leaves of the tree that are based on very little data and may not reflect reliable trends.
- A hypothesis, *h*, is said to overfit the training data is there exists another hypothesis which, *h´*, such that *h* has less error than *h´* on the training data but greater error on independent test data.



Avoid Overfitting in Classification

- The generated tree may overfit the training data
  - Too many branches, some may reflect anomalies due to noise or outliers
  - Result is in poor accuracy for unseen samples
- Two approaches to avoid overfitting
  - Prepruning: Halt tree construction early—do not split a node if this would result in the goodness measure falling below a threshold
    - Difficult to choose an appropriate threshold
  - Postpruning: Remove branches from a "fully grown" tree—get a sequence of progressively pruned trees
    - Use a set of data different from the training data to decide which is the "best pruned tree"

Additional Decision Tree Issues

- Better splitting criteria
  - Information gain prefers features with many values.
- Continuous features
- Predicting a real-valued function (regression trees)
- Missing feature values
- Features with costs
- Misclassification costs
- Incremental learning
  - ID4
  - ID5
- Mining large databases that do not fit in main memory

Naïve Bayes

- Uses bag of words model
- Relies on Bayesian rule for conditional probabilities

Bag of words

f(Cracks me up that this is so low rated. EXCELLENT version of Peter Rabbit - I liked the artistic license, it wasn't too far - it was well acted, well done digitally - and I loved it, my kids loved it AND my grandchildren, most importantly liked it! Thanks for the acting, the interpretation and making a "smart" movie.) = c

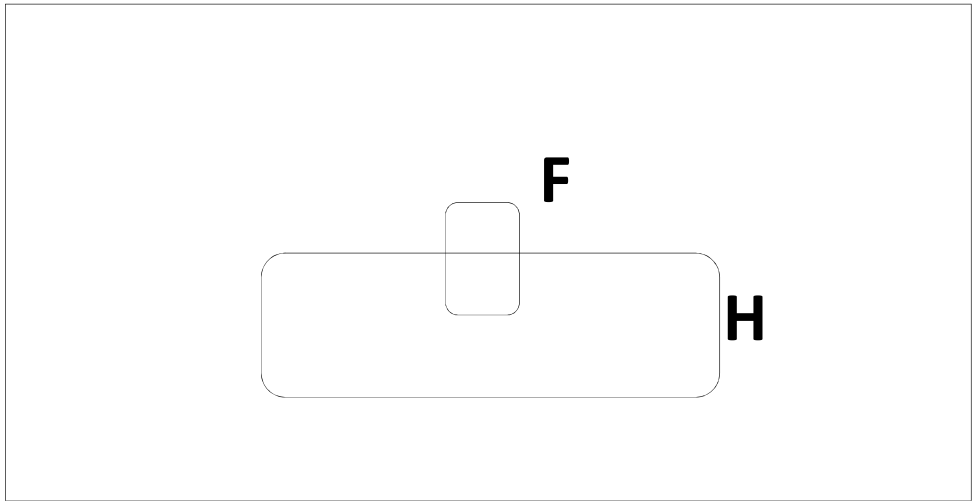
c = {pos, neg}

Look at all the words or a subset of words

BOW representation: does not care about the order

Conditional Probability

- P(A|B) = Fraction of worlds in which B is true that also have A true

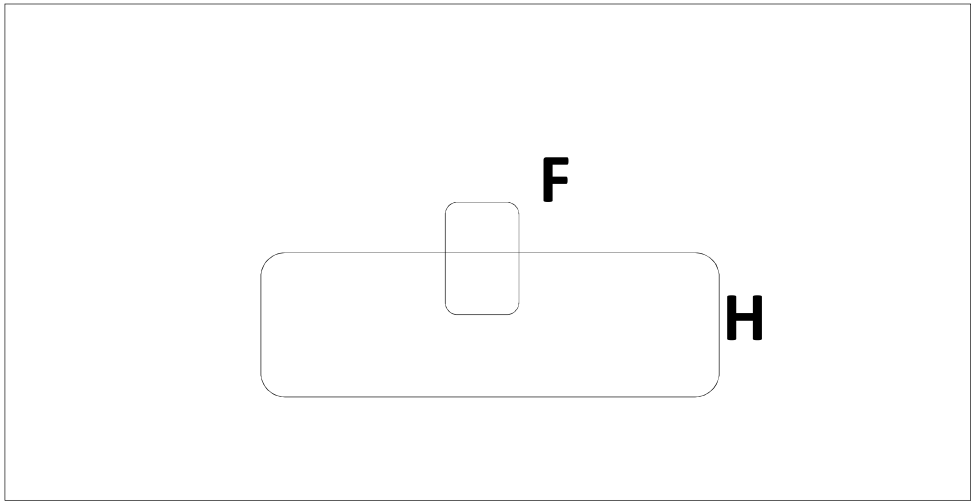


H = “Have a headache”
F = “Coming down with Flu”

P(H) = 1/10
P(F) = 1/40
P(H|F) = 1/2

“Headaches are rare and flu is rarer, but if you’re coming down with ‘flu there’s a 50-50 chance you’ll have a headache.”

Conditional Probability



H = “Have a headache”
F = “Coming down with Flu”

P(H) = 1/10
P(F) = 1/40
P(H|F) = 1/2

P(H|F) = Fraction of flu-inflicted worlds in which you have a headache

= #worlds with flu and headache

#worlds with flu

= Area of “H and F” region

Area of “F” region

= P(H and F)

P(F)

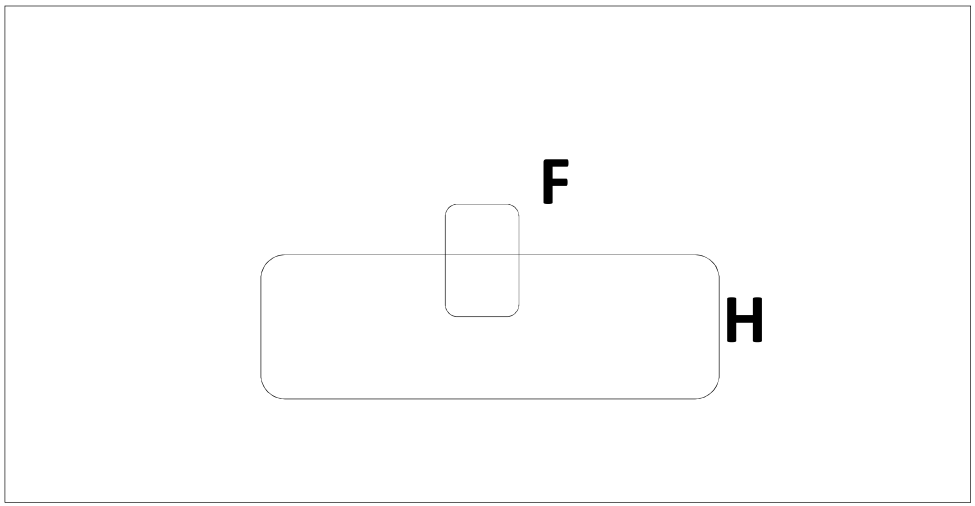
Definition of Conditional Probability

$$P(A|B) = \frac{P(A \text{ and } B)}{P(B)}$$

Corollary: The Chain Rule

$$P(A \text{ and } B) = P(A|B)\,P(B)$$

Probabilistic Inference



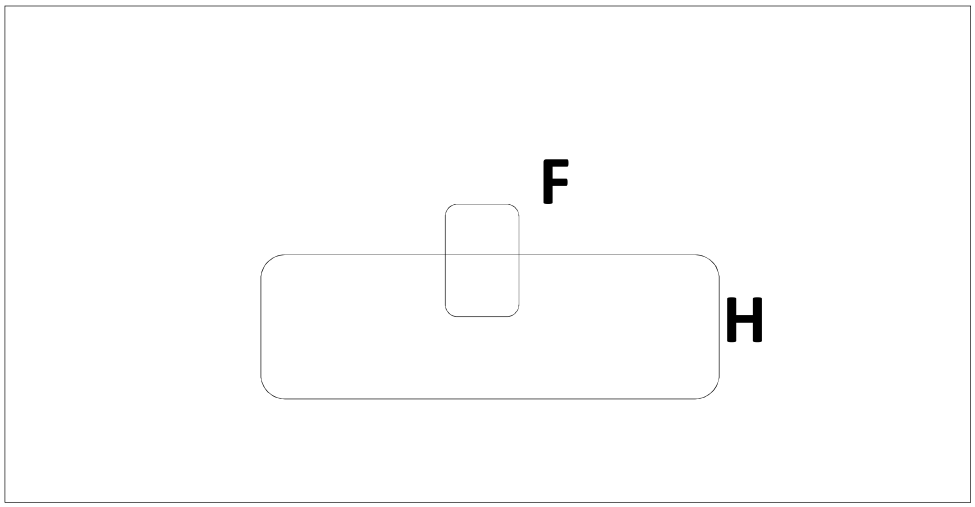
H = “Have a headache”
F = “Coming down with Flu”

P(H) = 1/10
P(F) = 1/40
P(H|F) = 1/2

One day you wake up with a headache. You think: “Drat! 50% of flus are associated with headaches so I must have a 50-50 chance of coming down with flu”

Is this reasoning good?

Probabilistic Inference



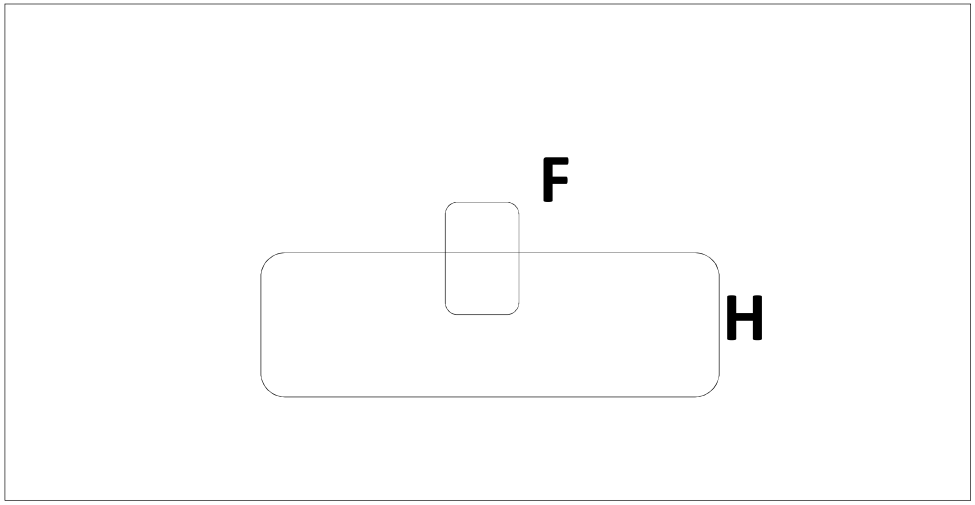
H = “Have a headache”
F = “Coming down with Flu”

P(H) = 1/10
P(F) = 1/40
P(H|F) = 1/2

P(F and H) = …

P(F|H) = …

Probabilistic Inference



H = “Have a headache”
F = “Coming down with Flu”

P(H) = 1/10
P(F) = 1/40
P(H|F) = 1/2

$$P(F \text{ and } H) = P(H \mid F) \times P(F) = \frac{1}{2} \times \frac{1}{40} = \frac{1}{80}$$

$$P(F \mid H) = \frac{P(F \text{ and } H)}{P(H)} = \frac{1/80}{1/10} = \frac{1}{8}$$

What we just did…

$$P(Y|X) = \frac{P(X \& Y)}{P(X)} = \frac{P(X|Y)\,P(y)}{P(X)}$$

This is Bayes Rule

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

d – document

c – class

Pick the best class

Bayes, Thomas (1763) An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London,* **53:370-418**

Interpretation of Bayes Rule

$$p(c|D) = \frac{p(D|c)p(c)}{p(D)}$$

- **Prior** $p(c)$: Information we have before observation.
- **Posterior** $p(c|D)$: The distribution of c after observing D
- **Likelihood** $p(D|c)$: The likelihood of observing D given c

Crime Scene Analogy

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

- X is a crime scene. Y is a person who may have committed the crime
  - P(Y|X) - look at the scene - who did it?
  - P(Y) - who had a motive? (Profiler)
  - P(X|Y) - could they have done it? (CSI - transportation, access to weapons, alibi)
- Some people might have great motives, but no means - you need both!

Naive Bayes Classification

$$p(c|D) = \frac{p(D|c)p(c)}{p(D)}$$

- Calculate the probability of a class *t* given a document representation *x*
- Naive Bayes assumes all features are conditionally independent given the class.

Naive Bayes Classification

a short derivation

$$C_{MAP} = arg\max_{c \in C} P(c|d) =$$

$$= arg\max_{c \in C} \frac{P(d|c)(P(c)}{P(d)}$$

$$= arg\max_{c \in C} P(d|c)P(c)$$

C_{MAP} is "maximum a posteriori" = most likely class

The most likely class likely class is the one that maximizes the product of two probabilities

Why can we drop the denominator?

Naive Bayes Classification

$$C_{MAP} = arg\max_{c \in C} P(d|c)P(c)$$

$$= arg\max_{c \in C} P(x_1, x_2, \ldots, x_n|c)P(c)$$

Document is represented as a set of features

How to compute the probability of the class?

Naive Bayes Classification

$$C_{MAP} = arg\max_{c \in C} P(d|c)P(c)$$

$$= arg\max_{c \in C} P(x_1, x_2, \dots, x_n|c)P(c)$$

Document is represented as a set of features

How to compute the probability of the class? – easy (distribution in a set)

$O(|X|^n \bullet |C|)$ parameters

Multinomial Naïve Bayes Independence Assumptions

$$P(x_1, x_2, \ldots, x_n|c)$$

- BOW: Position does not matter
- Conditional independence: assume the feature probabilities $\mathrm{P}(x_i|\ c_j)$ are independent given the class *c*
- **Both assumptions are incorrect simplifying assumptions**

$$P(x_1, x_2, \ldots, x_n|c) = P(x_1, |c) \bullet P(x_2\ |c) \bullet P(x_3\ |c) \bullet \ldots \bullet P(x_n\ |c)$$

Multinomial Naïve Bayes Classifier

$$C_{MAP} = arg\max_{c \in C} P(x_1, x_2, \ldots, x_n | c) P(c)$$

$$C_{NB} = arg\max_{c \in C} P(c_j) \prod_{x \in X} P(x | c)$$

Learning the Multinomial Naïve Bayes Model

- First attempt: maximum likelihood estimates: use the frequencies in the data

Example

HOT	LIGHT	SOFT	RED
COLD	HEAVY	SOFT	RED
HOT	HEAVY	FIRM	RED
HOT	LIGHT	FIRM	RED
COLD	LIGHT	SOFT	BLUE
COLD	HEAVY	SOFT	BLUE
HOT	HEAVY	FIRM	BLUE
HOT	LIGHT	FIRM	BLUE
HOT	HEAVY	FIRM	?????

$$t^* = \underset{t}{argmax}\, p(x_1|t)p(x_2|t) \dots p(x_n|t)p(t)$$

$$p(t = red) = 0.5$$

$$p(t = blue) = 0.5$$

Example

HOT	LIGHT	SOFT	RED
COLD	HEAVY	SOFT	RED
HOT	HEAVY	FIRM	RED
HOT	LIGHT	FIRM	RED
COLD	LIGHT	SOFT	BLUE
COLD	HEAVY	SOFT	BLUE
HOT	HEAVY	FIRM	BLUE
HOT	LIGHT	FIRM	BLUE
HOT	HEAVY	FIRM	?????

$$t^* = \underset{t}{argmax}\, p(x_1|t)p(x_2|t) \dots p(x_n|t)p(t)$$

$p(hot|t = red) = 0.75$ $p(heavy|t = red) = 0.75$ $p(firm|t = red) = 0.5$

$p(hot|t = blue) = 0.5$ $p(heavy|t = blue) = 0.5$ $p(firm|t = blue) = 0.5$

Example

HOT	LIGHT	SOFT	RED
COLD	HEAVY	SOFT	RED
HOT	HEAVY	FIRM	RED
HOT	LIGHT	FIRM	RED
COLD	LIGHT	SOFT	BLUE
COLD	HEAVY	SOFT	BLUE
HOT	HEAVY	FIRM	BLUE
HOT	LIGHT	FIRM	BLUE
HOT	HEAVY	FIRM	?????

$$t^* = \underset{t}{argmax}\, p(x_1|t)p(x_2|t) \dots p(x_n|t)p(t)$$

$$p(hot|t = red) * p(heavy|t = red) * p(firm|t = red) * p(c = red) = 0.75 * 0.5^3 = 0.09375$$

$$p(hot|t = blue) * p(heavy|t = blue) * p(firm|t = blue) * p(c = blue) = 0.5^4 = 0.0625$$

Problem

- If one of the likelihoods is 0 then we are never going to pick the corresponding class

- **Solution**: Laplace (add-1 smoothing)

Multinomial Naïve Bayes: Learning

- Extract *Vocabulary*
- Calculate $P(c_j)$: for each c_j

  $docs_j$: all docs in c_j

$$P(c_j) = \frac{|docs_j|}{|total\ \#\ documents|}$$

Multinomial Naïve Bayes: Learning

- Calculate $P(w_k \mid c_j)$: terms

  $Text_j$: single doc containing all the docs

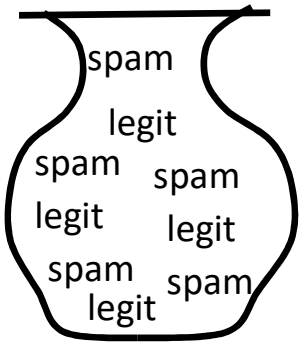
  For each w_k in Vocabulary, n_k : # of occurrences of w_k in T_j

$$P(w_k|c_j) = \frac{n_k + \alpha}{n + \alpha|Vocabulary|}$$

Unknown words?

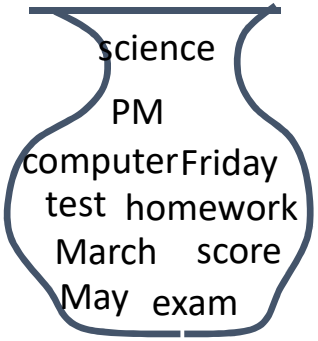
- Add one more word to the vocabulary
- NB is in close relationship to Language modelling

Naïve Bayes Generative Model for Text



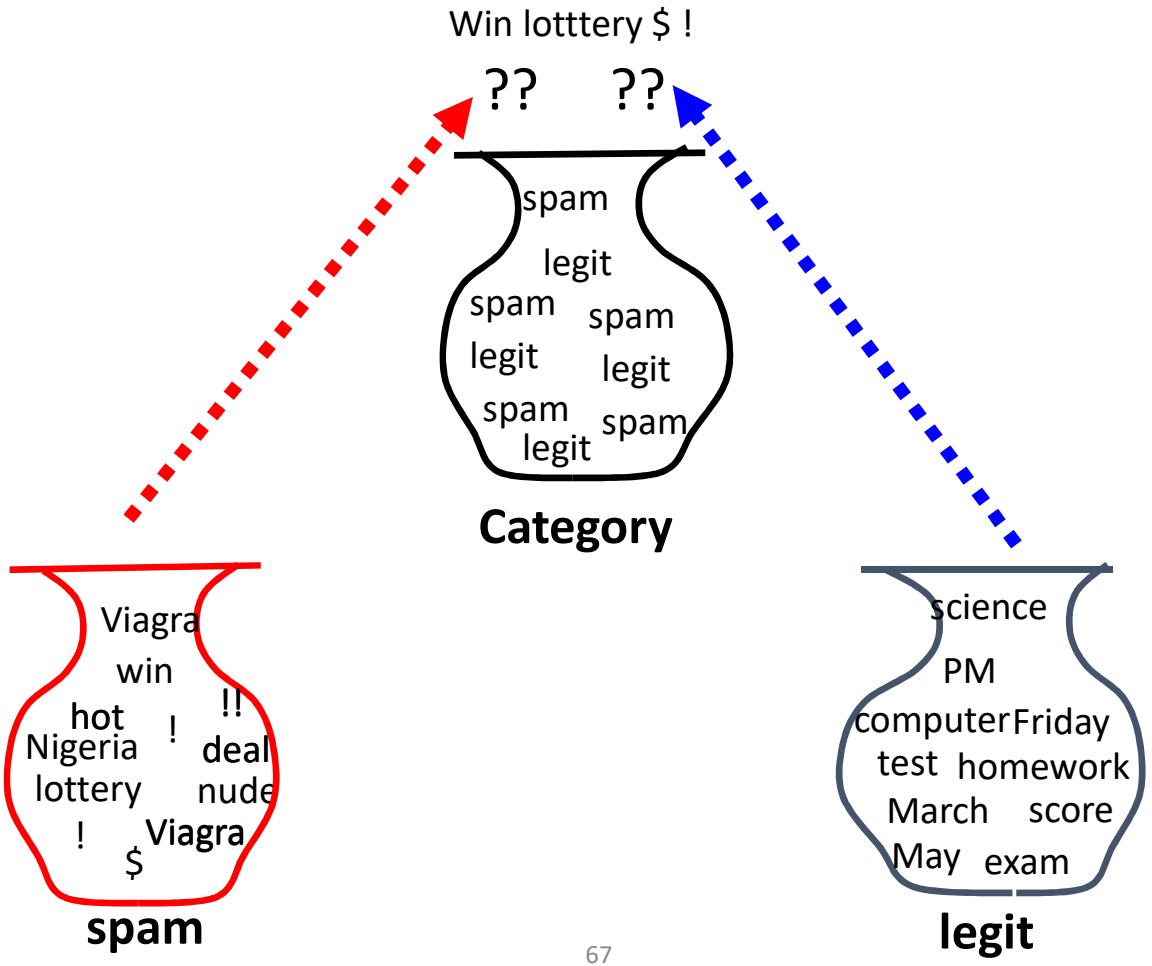
Category

spam



legit

Naïve Bayes Generative Model for Text



Underflow Prevention

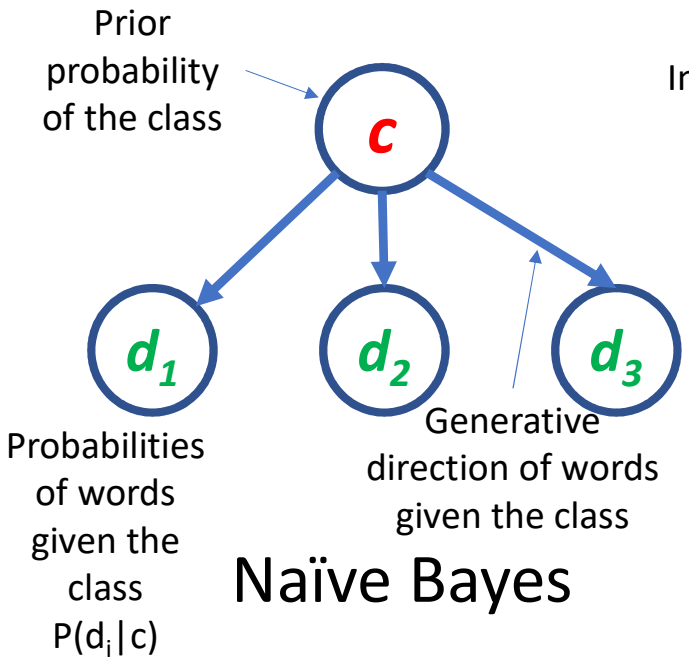
- Multiplying lots of probabilities, which are between 0 and 1 by definition, can result in floating-point underflow.
- Since $\log(xy) = \log(x) + \log(y)$, it is better to perform all computations by summing logs of probabilities rather than multiplying probabilities.
- Class with highest final un-normalized log probability score is still the most probable.

Naïve Bayes Posterior Probabilities

- Classification results of naïve Bayes (the class with maximum posterior probability) are usually fairly accurate.
- However, due to the inadequacy of the conditional independence assumption, the actual posterior-probability numerical estimates are not.
  - Output probabilities are generally very close to 0 or 1.

Logistic regression

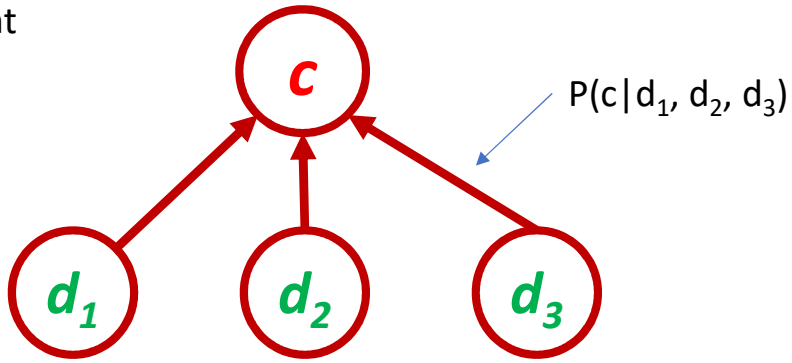
Generative vs. Discriminative Models



Naïve Bayes

Generative

In both cases we want to predict the class



Logistic Regression

Discriminative

Linear regression

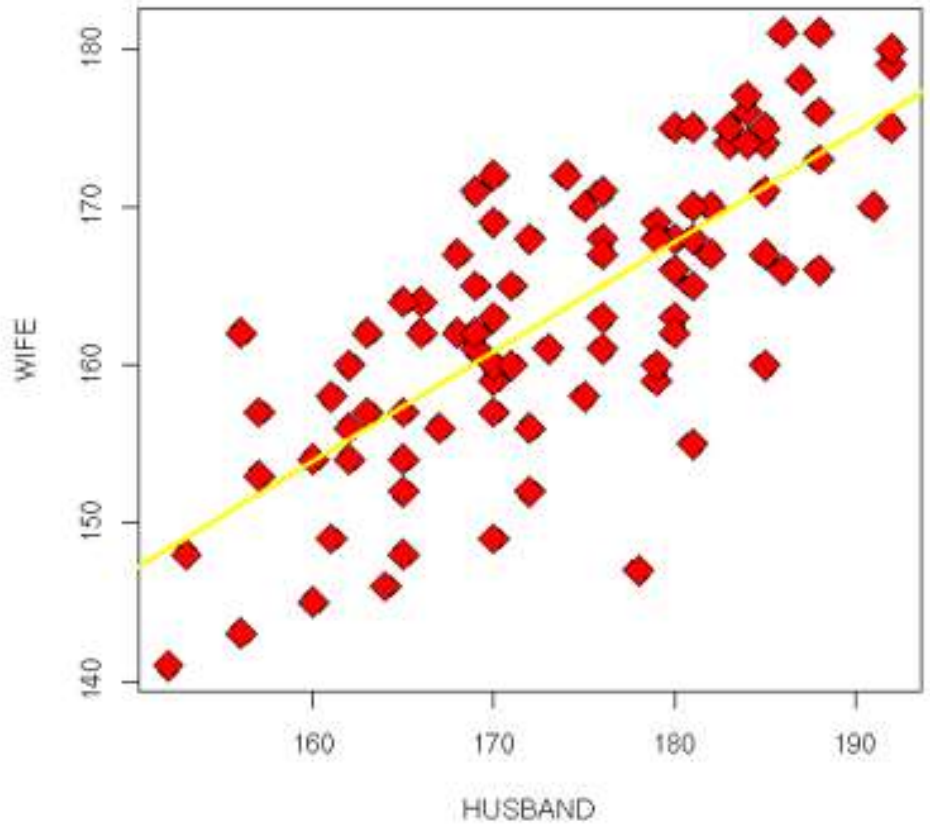
- What is a *regression* model?
  - A regression model is a model of the relationships between some covariates (predictors) and an outcome. Specifically, regression is a model of the average outcome given the covariates

- For height of couples data: a mathematical model, using only Husband's height:

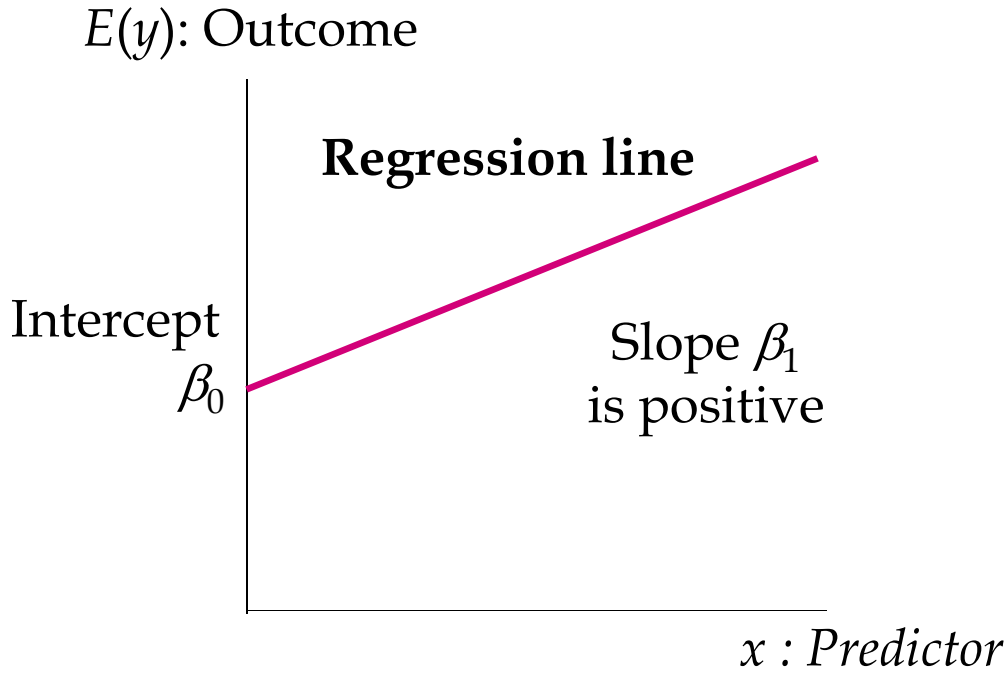
  $$Wife = f(Husband) + \varepsilon$$

- where f gives the average height of the wife of a man of height Husband and $\boldsymbol{\varepsilon}$ is the random error.

Height data



Simple Linear Regression Equation:



Logistic Regression: Definition

- Weight vector β_i
- Observations X_i
- "Bias" β_0 (like intercept in linear regression)

$$P(Y = 0|X) = \frac{1}{1 + \exp[\beta_o + \sum_\iota \beta_\iota \mathrm{X}_\iota]}$$

$$P(Y = 1|X) = \frac{[\beta_o + \sum_\iota \beta_\iota \mathrm{X}_\iota]}{1 + \exp[\beta_o + \sum_\iota \beta_\iota \mathrm{X}_\iota]}$$

For shorthand, we'll say that:

$$P(Y = 0|X) = \sigma(-(\beta_o + \sum_\iota \beta_\iota \mathrm{X}_\iota))$$

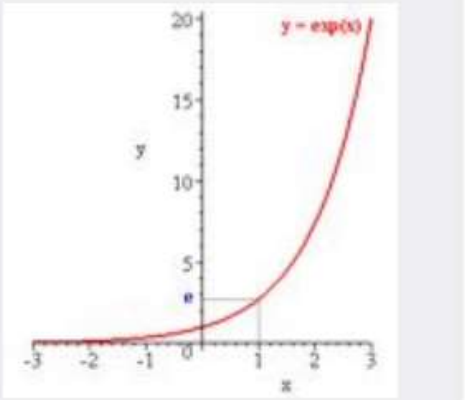
$$P(Y = 1|X) = 1 - \sigma(-(\beta_o + \sum_\iota \beta_\iota \mathrm{X}_\iota))$$

Where

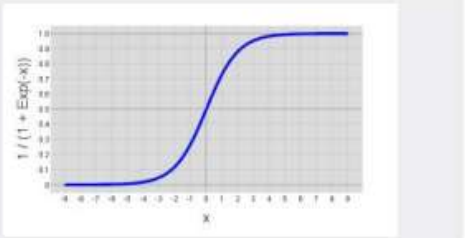
$$\sigma(\mathrm{z}) = \frac{1}{1 + \exp[-z]}$$

What's this "exp"?

Exponential



Logistic



- exp[x] is shorthand for e^x
- e is a special number, about 2.71828
  - e^x is the limit of compound interest formula as compounds become infinitely small
  - It's the function whose derivative is itself
- The "logistic" function is $\sigma(z) = \frac{1}{1+e^{-z}}$
- Looks like an "S"
- Always between 0 and 1.
  - Allows us to model probabilities
  - Different from **linear** regression

Logistic Regression Example

feature	coefficient	weight
bias	β_0	0.1
"viagra"	β_1	2.0
"mother"	β_2	−1.0
"work"	β_3	−0.5
"nigeria"	β_4	3.0

- What does $Y = 1$ mean?

Example 1: Empty document?
X = {}

- $P(Y = 0) = \frac{1}{1+\mathrm{e}^{[0.1]}} = 0.48$
- $P(Y = 1) = \frac{\exp[0.1]}{1+\exp[0.1]} = 0.52$
- Bias β_0 encodes the prior probability of a class

Logistic Regression Example

feature	coefficient	weight
bias	β_0	0.1
"viagra"	β_1	2.0
"mother"	β_2	−1.0
"work"	β_3	−0.5
"nigeria"	β_4	3.0

- What does $Y = 1$ mean?

Example 1: Empty document?
X = {Mother, Nigeria}

- $P(Y = 0) = \frac{1}{1+\exp[0.1-1.0+3.0]} = 0.11$
- $P(Y = 1) = \frac{\exp[0.1-1.0+3.0]}{1+\exp[0.1-1.0+3.0]} = 0.88$
- Include bias, and sum the other weights