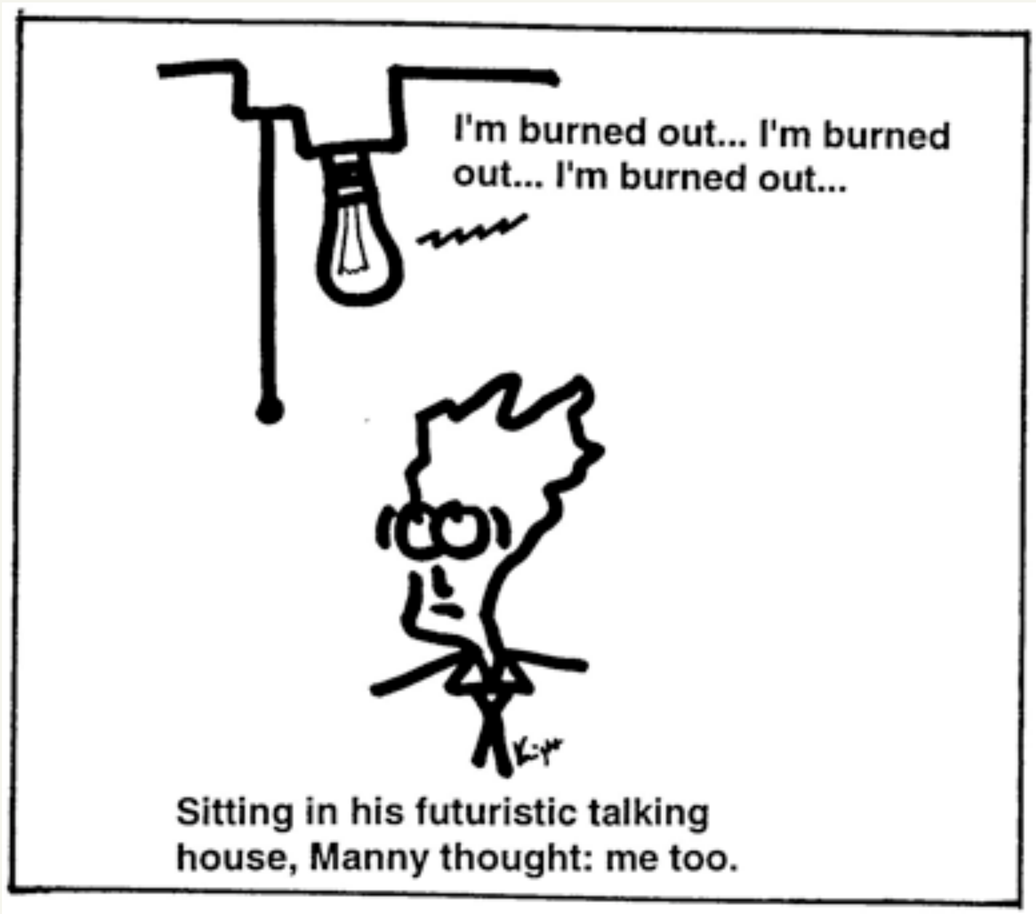


http://www.isi.edu/natural-language/people/knight3.html

Text Classification

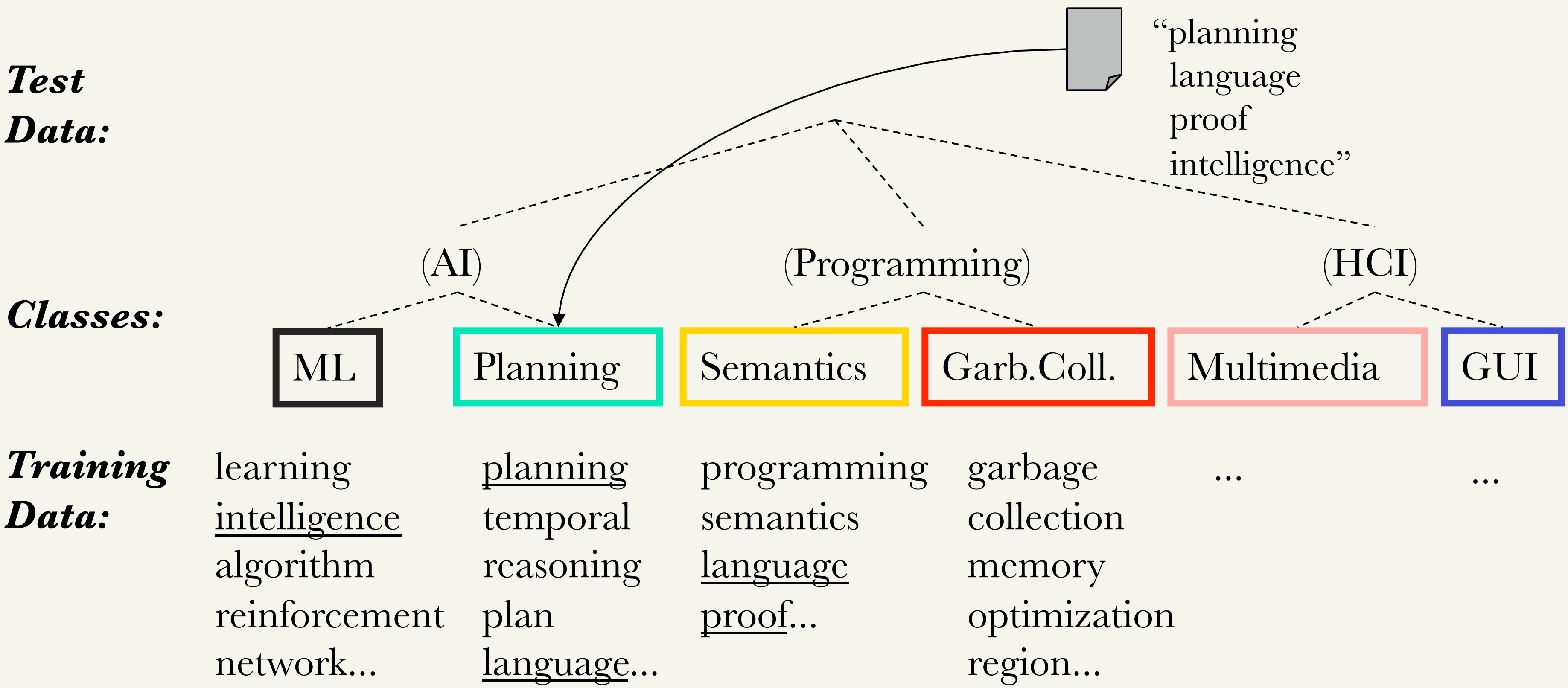
David Kauchak
cs160
Fall 2009

adapted from:
http://www.stanford.edu/class/cs276/handouts/lecture10-textcat-naivebayes.ppt
http://www.stanford.edu/class/cs276/handouts/lecture11-vector-classify.ppt
http://www.stanford.edu/class/cs276/handouts/lecture12-SVMs.ppt

Administrative

- Colloquium
- Project proposals
  - year and where published for references
  - speed is important for most of your approaches
- Project status report due 11/18
  - be specific!
  - but, be concise

Document Classification



How might this be useful for IR?

Standing queries

- The path from information retrieval to text classification:
  - You have an information need, say:
    - Unrest in the Niger delta region
  - You want to rerun an appropriate query periodically to find new news items on this topic
  - You will be sent new documents that are found
    - I.e., it's classification not ranking
- Such queries are called ***standing queries***
  - Long used by "information professionals"
  - A modern mass instantiation is **Google Alerts**

Spam filtering

From: "" <takworlld@hotmail.com>
Subject: real estate is the only way... gem oalvgkay

Anyone can buy real estate with no money down

Stop paying rent TODAY !

There is no need to spend hundreds or even thousands for similar courses

I am 22 years old and I have already purchased 6 properties using the methods outlined in this truly INCREDIBLE ebook.

Change your life NOW !

===
Click Below to order:
http://www.wholesaledaily.com/sales/nmd.htm
===

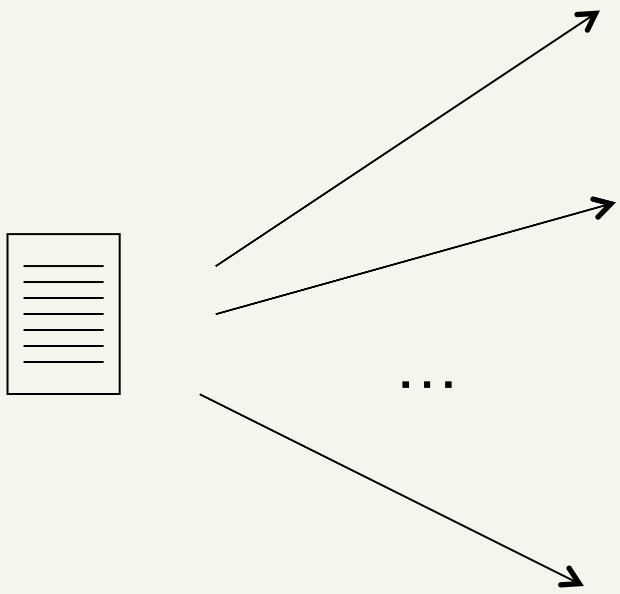
More Text Classification Examples:

Many search engine functionalities use classification

Assign labels to each document or web-page:

- Labels are most often topics such as Yahoo-categories
  e.g., "finance," "sports," "news>world>asia>business"
- Labels may be genres
  e.g., "editorials" "movie-reviews" "news“
- Labels may be opinion on a person/product
  e.g., “like”, “hate”, “neutral”
- Labels may be domain-specific
  e.g., "interesting-to-me" : "not-interesting-to-me”
  e.g., “contains adult language” : “doesn’t”
  e.g., language identification: English, French, Chinese, …
  e.g., search vertical: about Linux versus not
  e.g., “link spam” : “not link spam”

How would you do it?



Pros and cons of different approaches?

Manual approach

- Manual classification
  - Used by Yahoo! (originally; now present but downplayed), Looksmart, about.com, ODP, PubMed
  - Very accurate when job is done by experts
  - Consistent when the problem size and team is small
  - Difficult and expensive to scale
    - Means we need automatic classification methods for big problems

A slightly better manual approach

- Hand-coded rule-based systems
  - One technique used by many spam filter, Reuters, CIA, etc.
  - Companies (Verity) provide “IDE” for writing such rules
  - E.g., assign category if document contains a given boolean combination of words
  - Accuracy is often very high if a rule has been carefully refined over time by a subject expert
  - Building and maintaining these rules is expensive

A Verity topic (a complex classification rule)

```
comment line                  # Beginning of art topic definition
top-level topic               art ACCRUE
                                  /author = "fsmith"
topic definition modifiers        /date   = "30-Dec-01"
                                  /annotation = "Topic created
                                                by fsmith"
 subtopictopic                * 0.70 performing-arts ACCRUE
   evidencetopic              ** 0.50 WORD
   topic definition modifier      /wordtext = ballet
   evidencetopic              ** 0.50 STEM
   topic definition modifier      /wordtext = dance
   evidencetopic              ** 0.50 WORD
   topic definition modifier      /wordtext = opera
   evidencetopic              ** 0.30 WORD
   topic definition modifier      /wordtext = symphony
subtopic                      * 0.70 visual-arts ACCRUE
                              ** 0.50 WORD
                                  /wordtext = painting
                              ** 0.50 WORD
                                  /wordtext = sculpture
subtopic                      * 0.70 film ACCRUE
                              ** 0.50 STEM
                                  /wordtext = film
subtopic                      ** 0.50 motion-picture PHRASE
                              *** 1.00 WORD
                                  /wordtext = motion
                              *** 1.00 WORD
                                  /wordtext = picture
                              ** 0.50 STEM
                                  /wordtext = movie
subtopic                      * 0.50 video ACCRUE
                              ** 0.50 STEM
                                  /wordtext = video
                              ** 0.50 STEM
                                  /wordtext = vcr
                              # End of art topic
```

- Note:
  - maintenance issues (author, etc.)
  - Hand-weighting of terms

Automated approaches

- Supervised learning of a document-label assignment function
  - Many systems partly rely on machine learning (Autonomy, MSN, Verity, Enkata, Yahoo!, …)
    - k-Nearest Neighbors (simple, powerful)
    - Naive Bayes (simple, common method)
    - Support-vector machines (new, more powerful)
    - … plus many other methods
    - No free lunch: requires hand-classified training data
- Note that many commercial systems use a mixture of methods

Bayes' Rule

$$P(C,D) = P(C \mid D)P(D) = P(D \mid C)P(C)$$

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

How can we use this?

Bayes' Rule

$$P(Class \mid Document) = \frac{\overbrace{P(D \mid C)}^{\text{conditional probability}}\overbrace{P(C)}^{\text{prior probability}}}{P(D)}$$

Naive Bayes Classifiers

Represent a document *D* based on a attribute values

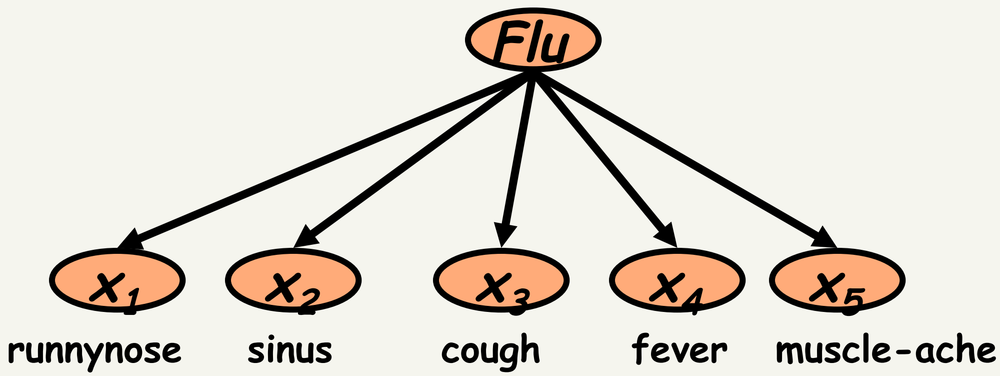
$$D = \langle x_1, x_2, \ldots, x_n \rangle$$

$$class = \underset{c_j \in C}{\operatorname{argmax}} P(c_j \mid x_1, x_2, \ldots, x_n)$$

$$= \underset{c_j \in C}{\operatorname{argmax}} \frac{P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)}{P(x_1, x_2, \ldots, x_n)}$$

$$= \underset{c_j \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)$$

The Naive Bayes Classifier



- **Conditional Independence Assumption:** features detect term presence and are independent of each other given the class:

$$P(x_1, \ldots, x_5 \mid C) = P(x_1 \mid C) \bullet P(x_2 \mid C) \bullet \cdots \bullet P(x_5 \mid C)$$

Estimating parameters

- I flip a coin 1000 times, how would you estimate the probability of heads?

- I roll a 6-sided die 1000 times, how you estimate the probability of getting a ‘6’?

For us:

$$class = \underset{c_j \in C}{\operatorname{argmax}} P(x_1 \mid c_j) P(x_2 \mid c_j) \ldots P(x_n \mid c_j) P(c_j)$$

Ideas?

Maximum likelihood estimates

$$\hat{P}(c_j) = \frac{N(C = c_j)}{N}$$

$$\frac{\text{number of document with class}}{\text{total number of document}}$$

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j)}{N(C = c_j)}$$

$$\frac{\text{number of document in class with feature}}{\text{number of document with class}}$$

What's the problem with this approach?

Problem with Max Likelihood

- What if we have seen no training cases where patient had no flu and muscle aches?

$$\hat{P}(X_5 = t \mid C = nf) = \frac{N(X_5 = t, C = nf)}{N(C = nf)} = 0$$

- Zero probabilities cannot be conditioned away, no matter the other evidence!

$$\ell = \arg\max_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c)$$

Smoothing to Avoid Overfitting

Make every event a little probable…

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j) + \lambda}{N(C = c_j) + k\lambda}$$

# of values of X_i

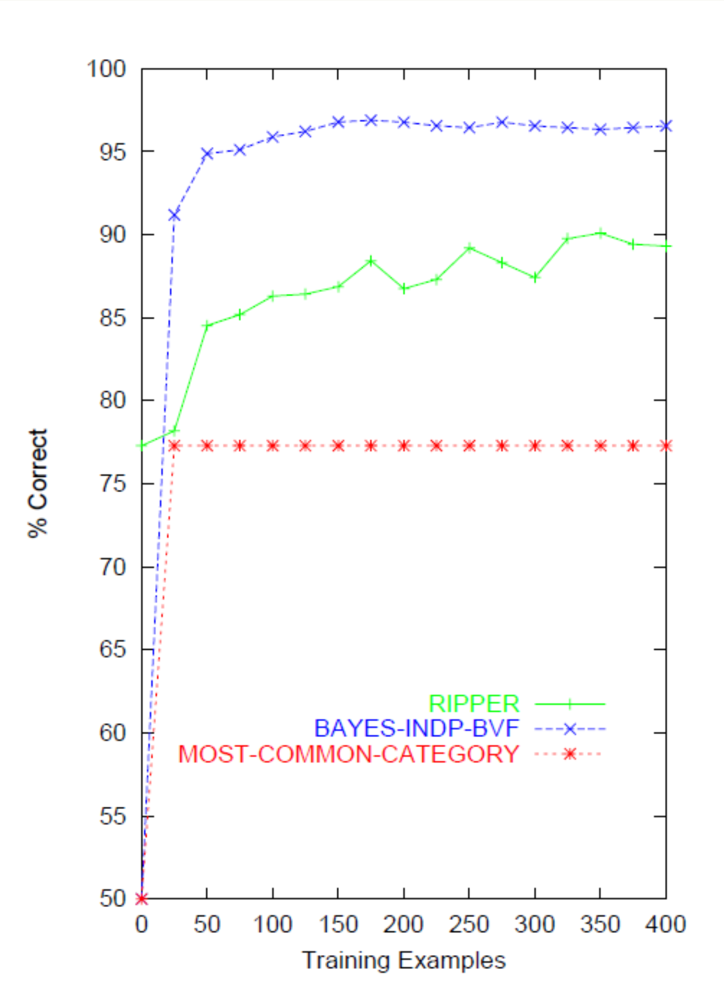
WebKB Experiment (1998)

- Classify webpages from CS departments into:
  - student, faculty, course,project
- Train on ~5,000 hand-labeled web pages
  - Cornell, Washington, U.Texas, Wisconsin
- Crawl and classify a new site (CMU)

- Results:

	Student	Faculty	Person	Project	Course	Departmt
Extracted	180	66	246	99	28	1
Correct	130	28	194	72	25	1
Accuracy:	72%	42%	79%	73%	89%	100%

Naive Bayes on spam email



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

SpamAssassin

- Naive Bayes has found a home in spam filtering
  - Paul Graham's *A Plan for Spam*
    - A mutant with more mutant offspring...
  - Naive Bayes-like classifier with weird parameter estimation
  - Widely used in spam filters
  - But also many other things: black hole lists, etc.

- Many email topic filters also use NB classifiers

NB: The good and the bad

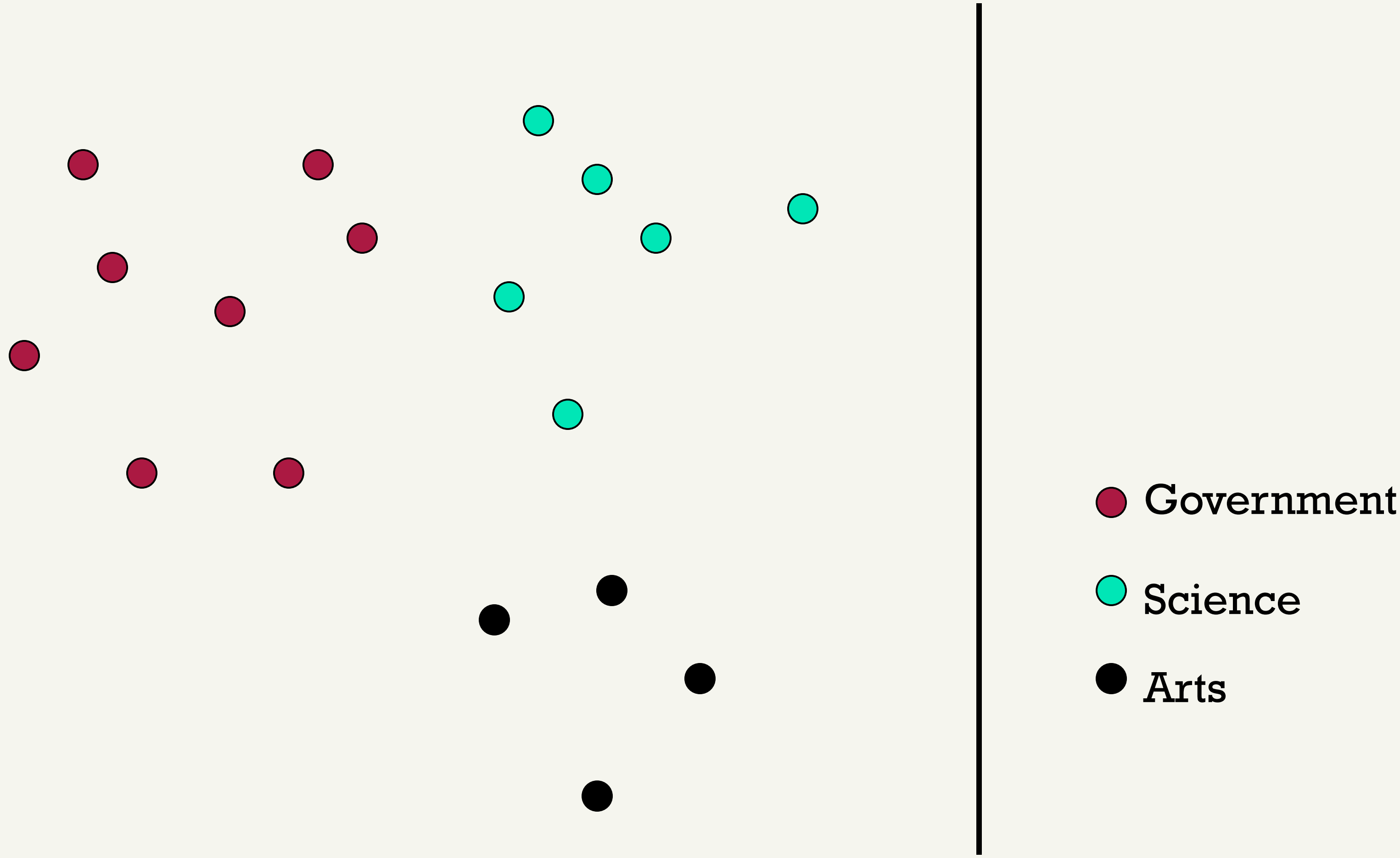
- Good
  - Easy to understand
  - Fast to train
  - Reasonable performance
- Bad
  - We can do better
  - Independence assumptions are rarely true
  - Smoothing is challenging
  - Feature selection is usually required

Recall: Vector Space Representation

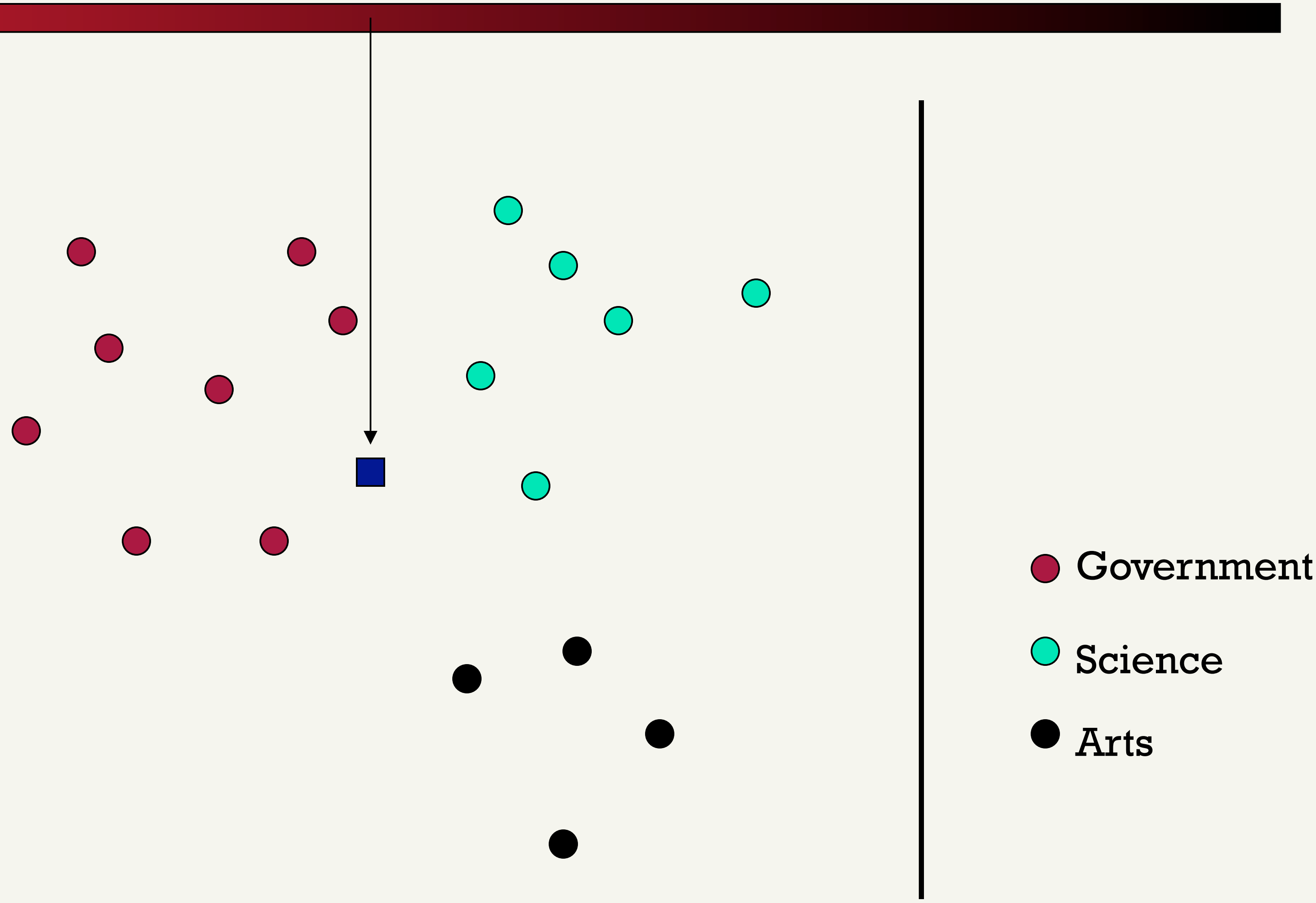
- Each document is a vector, one component for each term/word
- Normally normalize vectors to unit length
- High-dimensional vector space:
  - Terms are axes
  - 10,000+ dimensions, or even 100,000+
  - Docs are vectors in this space

- How can we do classification in this space?

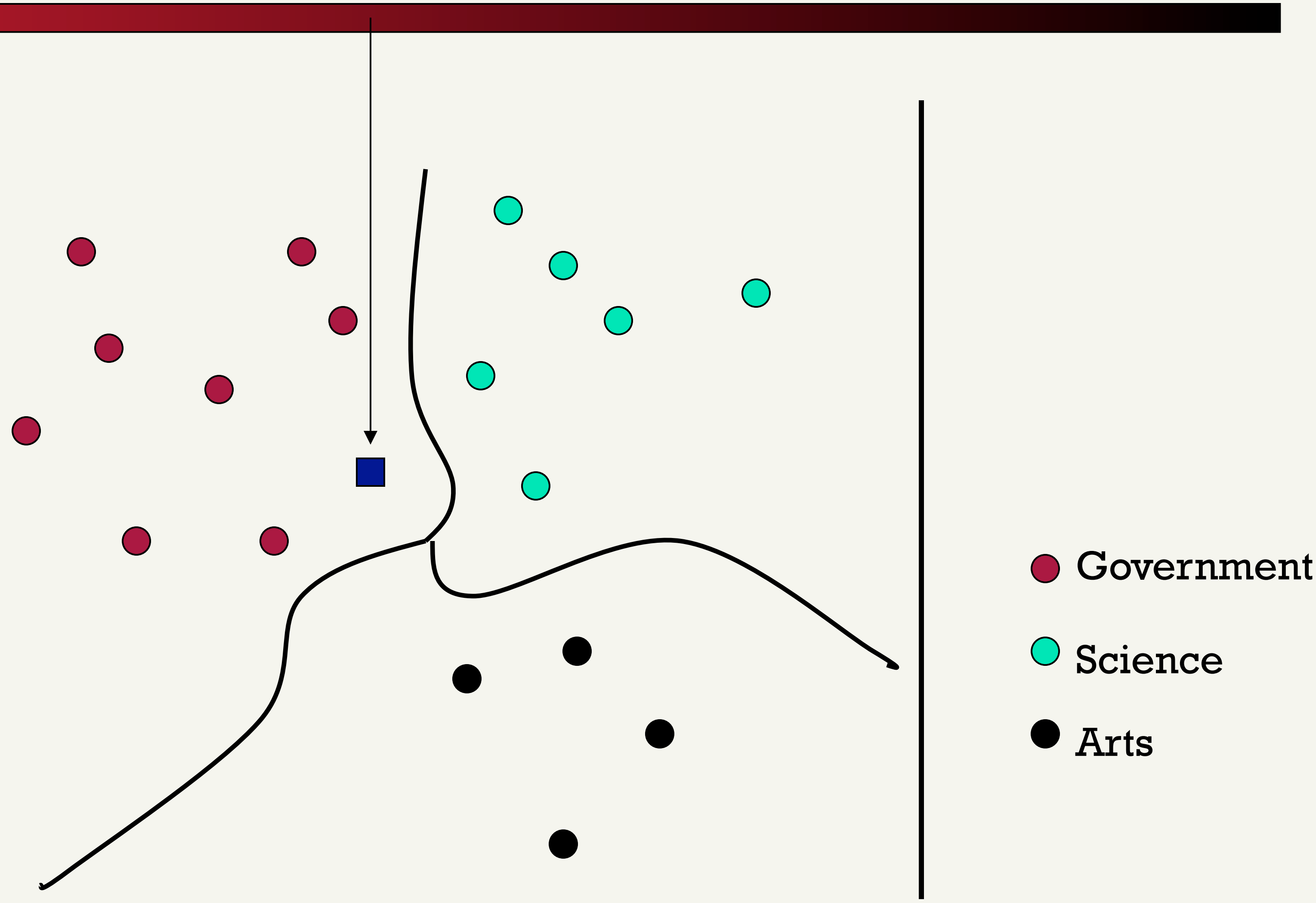
Documents in a Vector Space



Test Document of what class?



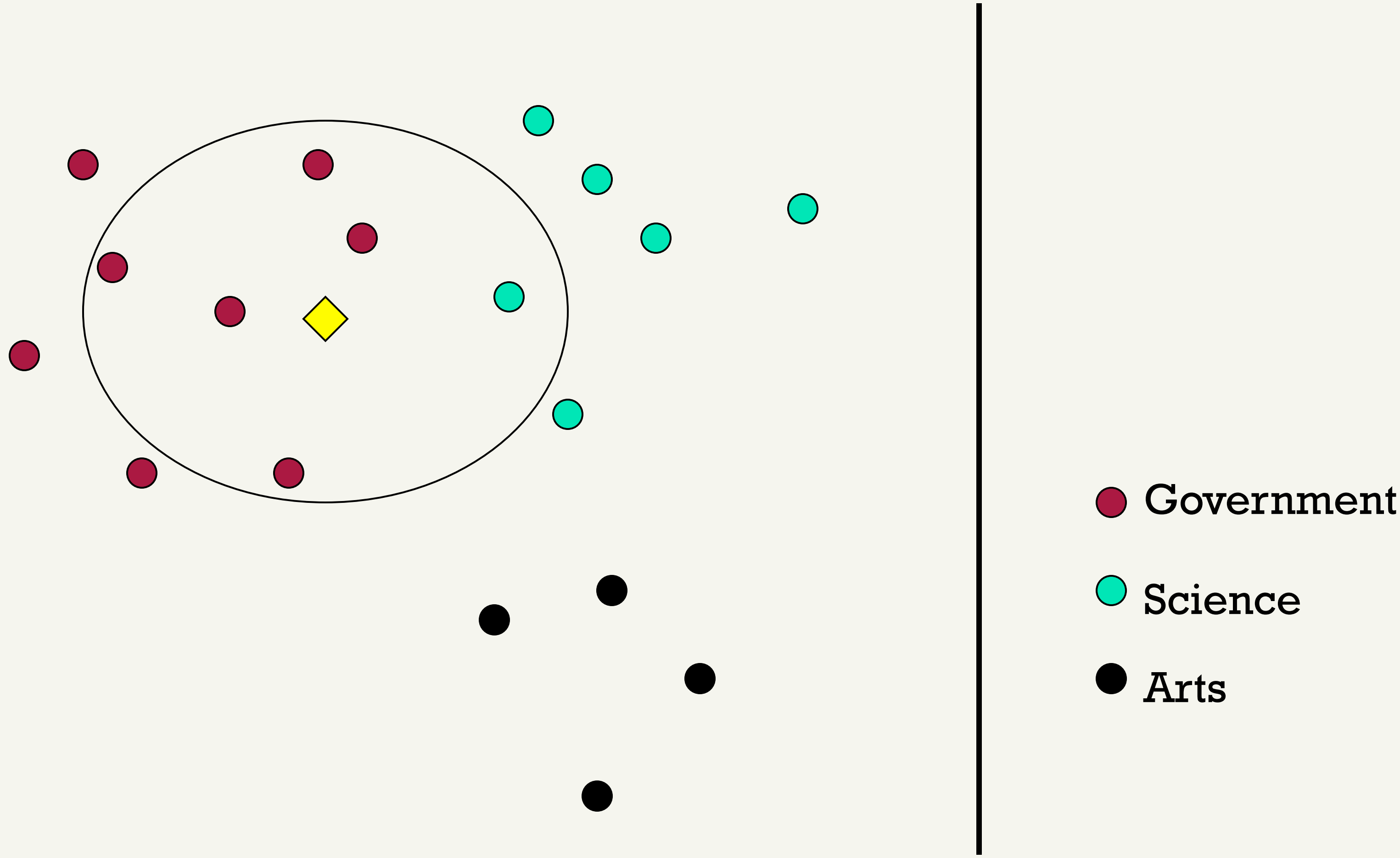
Test Document = Government



k-Nearest Neighbor (k-NN)

- To classify document ***d***:
  - Find ***k*** nearest neighbors of ***d***
  - Choose as the class the majority class within the ***k*** nearest neightbors
- Can get rough approximations of probability of belonging to a class as fraction of ***k***
- Does not explicitly compute boundary or model
- Also called:
  - Case-based learning
  - Memory-based learning
  - Lazy learning

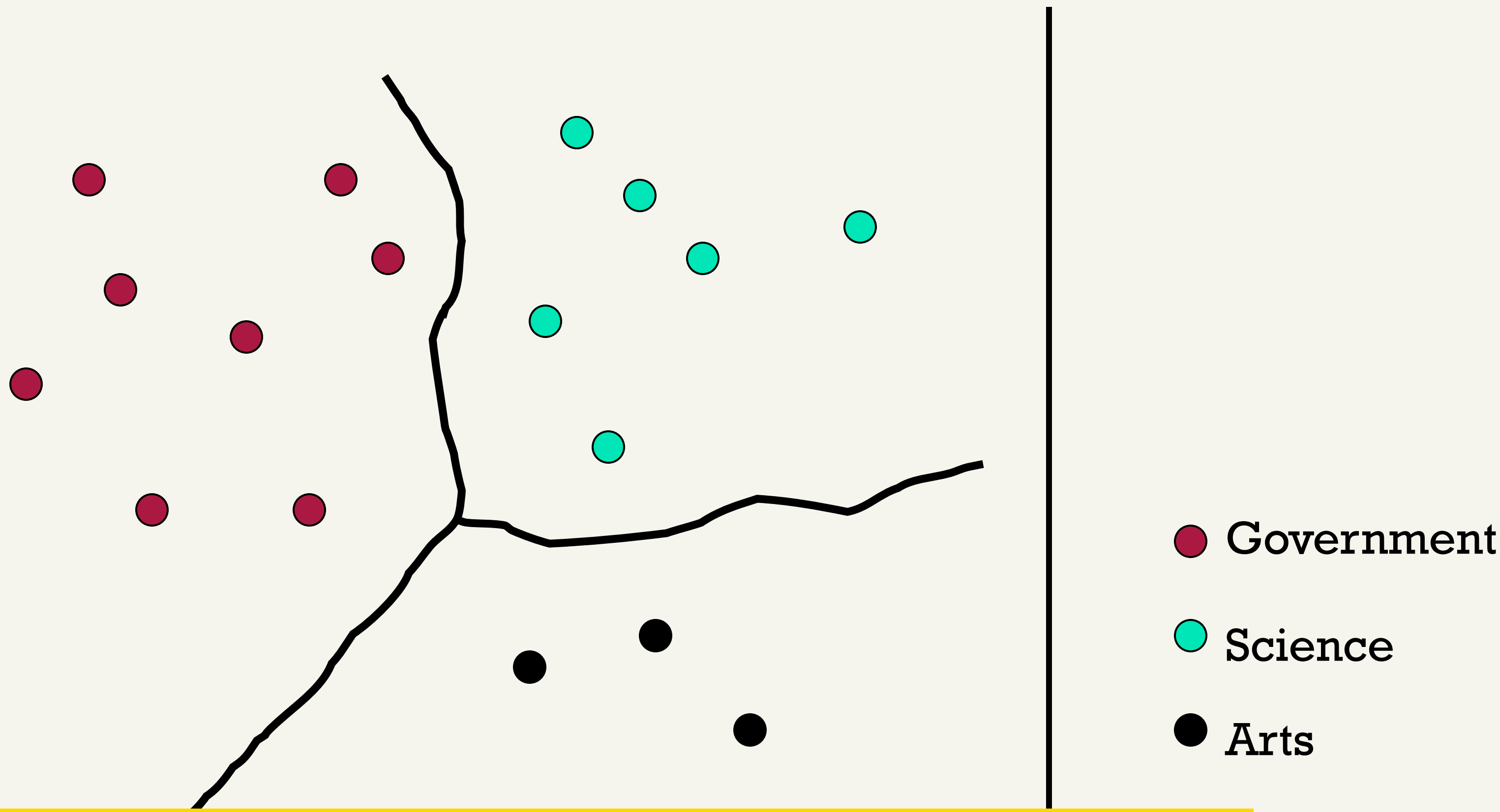
Example: k=6 (6-NN)



k Nearest Neighbor

- What value of k should we use?
  - Using only the closest example (1NN) to determine the class is subject to errors due to:
    - A single atypical example
    - Noise
  - Pick k too large and you end up with looking at neighbors that are not that close
  - Value of *k* is typically odd to avoid ties; 3 and 5 are most common.

k-NN decision boundaries



k-NN gives locally defined decision boundaries between classes – far away points do not influence each classification decision (unlike in Naïve Bayes, etc.)

Similarity Metrics

- Nearest neighbor methods depends on a similarity (or distance) metric
  - *Euclidean distance*.
  - Binary instance space is *Hamming distance* (number of feature values that differ)
  - For text, cosine similarity of tf.idf weighted vectors is typically most effective

k-NN: The good and the bad

- Good
  - No training is necessary
  - No feature selection necessary
  - Scales well with large number of classes
    - Don't need to train *n* classifiers for *n* classes
- Bad
  - Classes can influence each other
    - Small changes to one class can have ripple effect
  - Scores can be hard to convert to probabilities
  - Can be more expensive at test time
  - "Model" is all of your training examples which can be large

Bias/variance trade-off

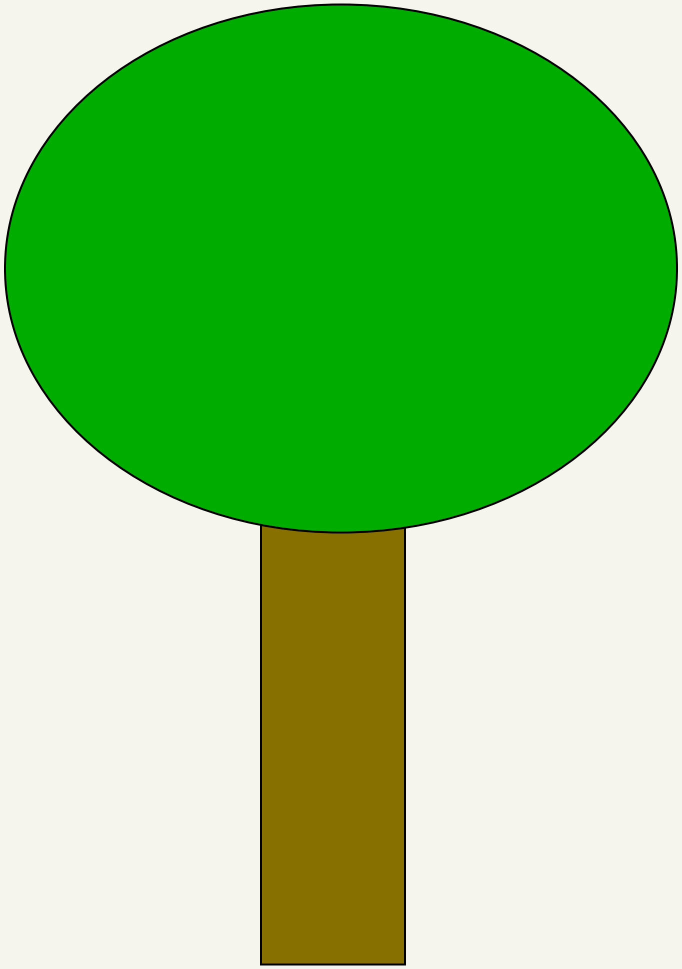
Is this a tree?

Bias/variance trade-off

Is this a tree?

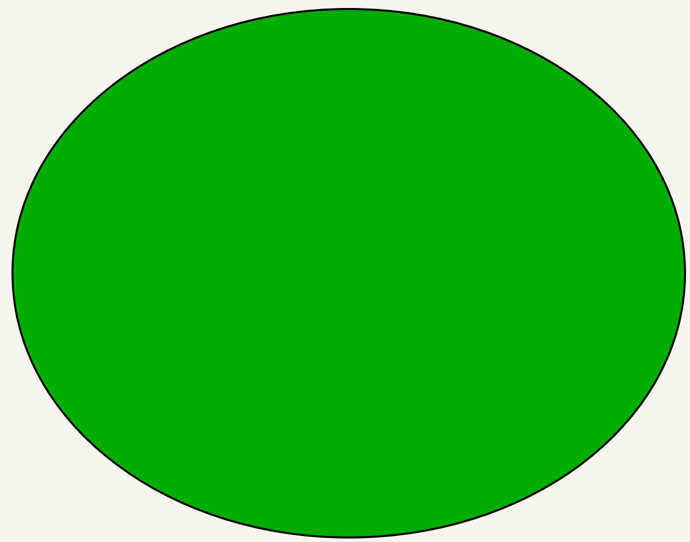
Bias/variance trade-off

Is this a tree?



Bias/variance trade-off

Is this a tree?



Bias vs. variance

- Another way to think about it:
  - Generalizability vs. Precision

- Consider asking a botanist: Is an object a tree?
  - High variance, low bias
    - Botanist who memorizes
    - Will always say “no” to new object (e.g., different # of leaves)
  - Low variance, high bias
    - Lazy botanist
    - Says “yes” if the object is green
  - You want the middle ground

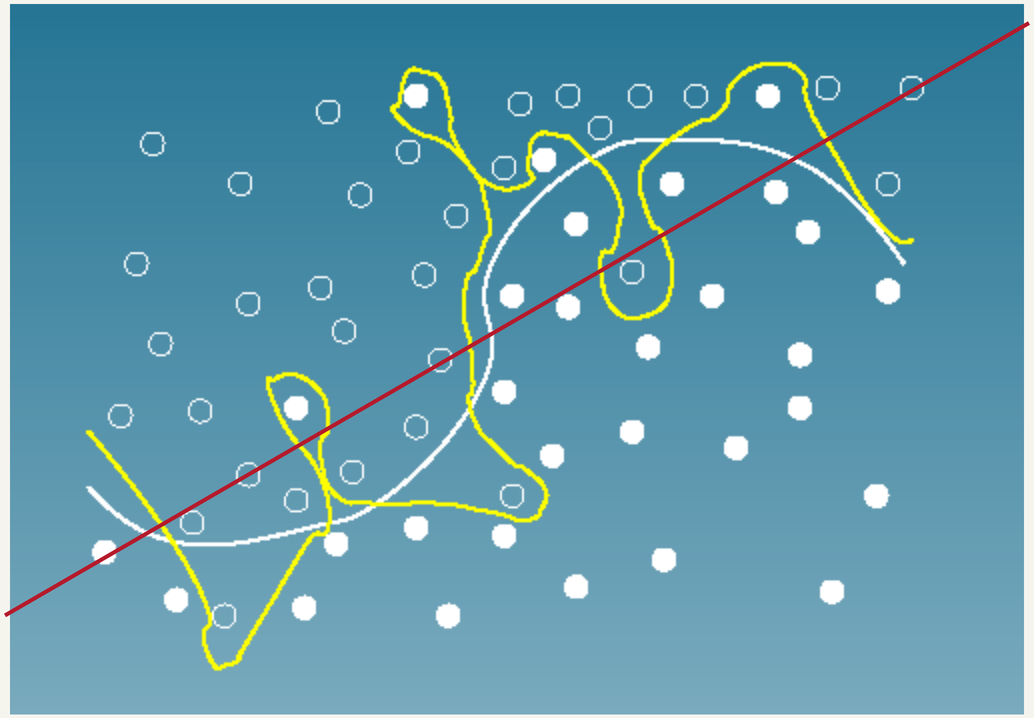
(Example due to C. Burges)

k-NN vs. Naive Bayes

How do k-NN and NB sit on the variance/bias plane?

- k-NN has high variance and low bias.
  - Infinite memory
- NB has low variance and high bias.
  - Decision surface has to be linear (hyperplane – see later)

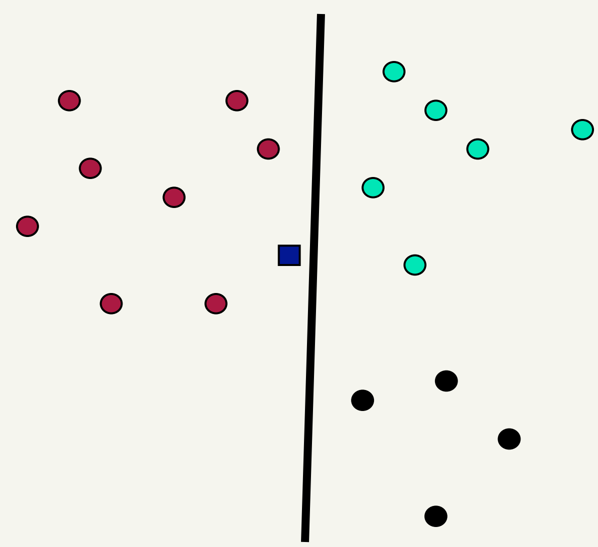
Bias vs. variance: Choosing the correct model capacity



Which separating line should we use?

Separation by Hyperplanes

- A strong high-bias assumption is *linear separability*:
  - in 2 dimensions, can separate classes by a line
  - in higher dimensions, need hyperplanes



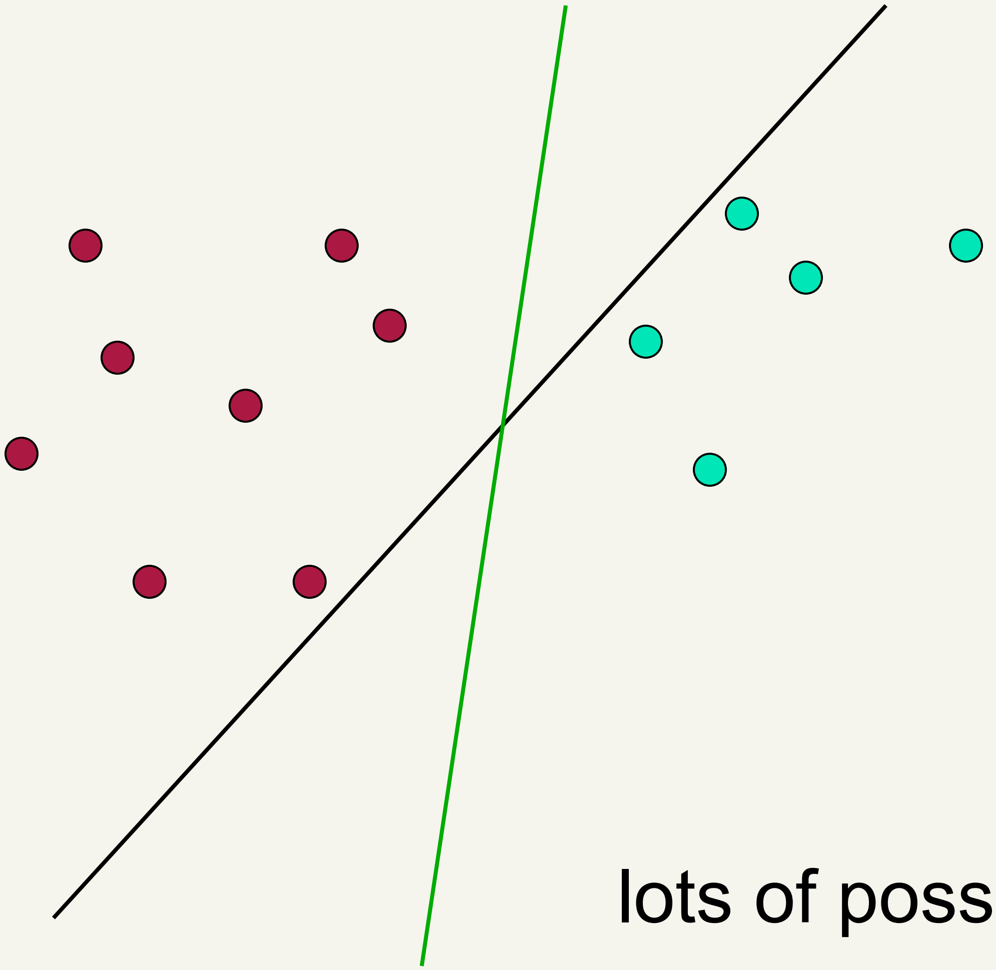
Lots of linear classifiers

- Many common text classifiers are linear classifiers
  - Naïve Bayes
  - Perceptron
  - Rocchio
  - Logistic regression
  - Support vector machines (with linear kernel)
  - Linear regression

- Despite this similarity, noticeable performance difference

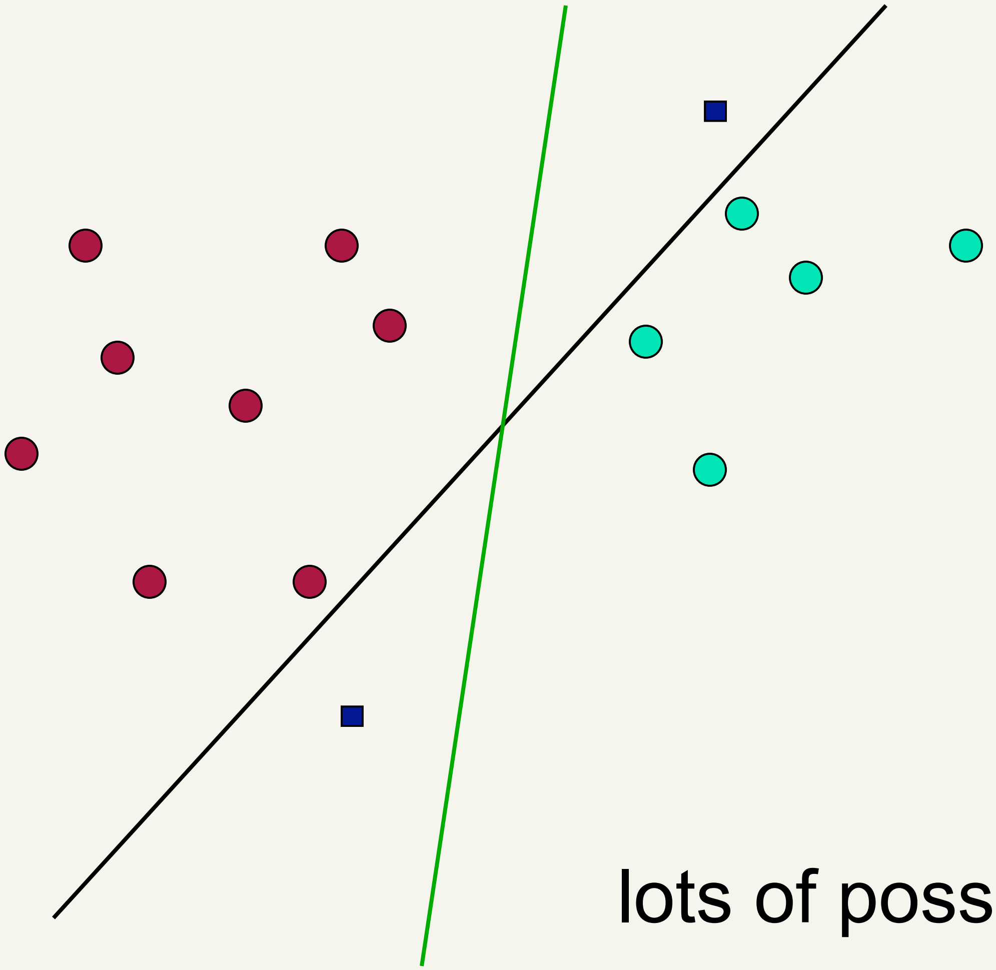
How might algorithms differ?

Which Hyperplane?



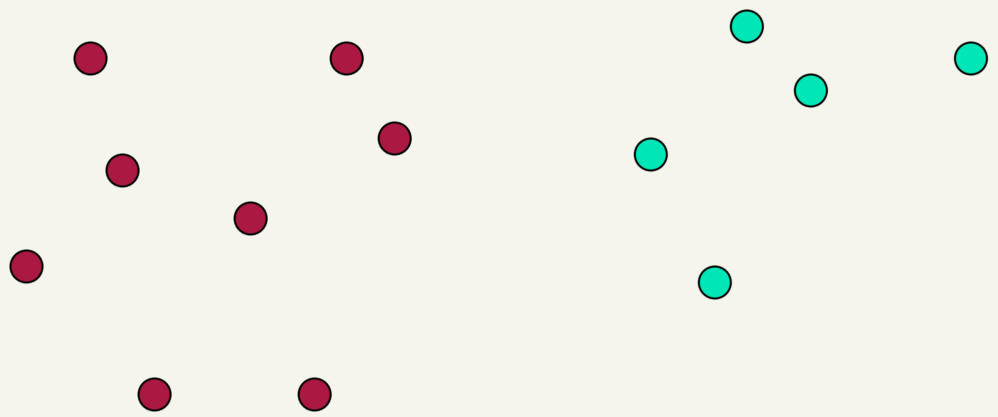
lots of possible solutions

Which Hyperplane?

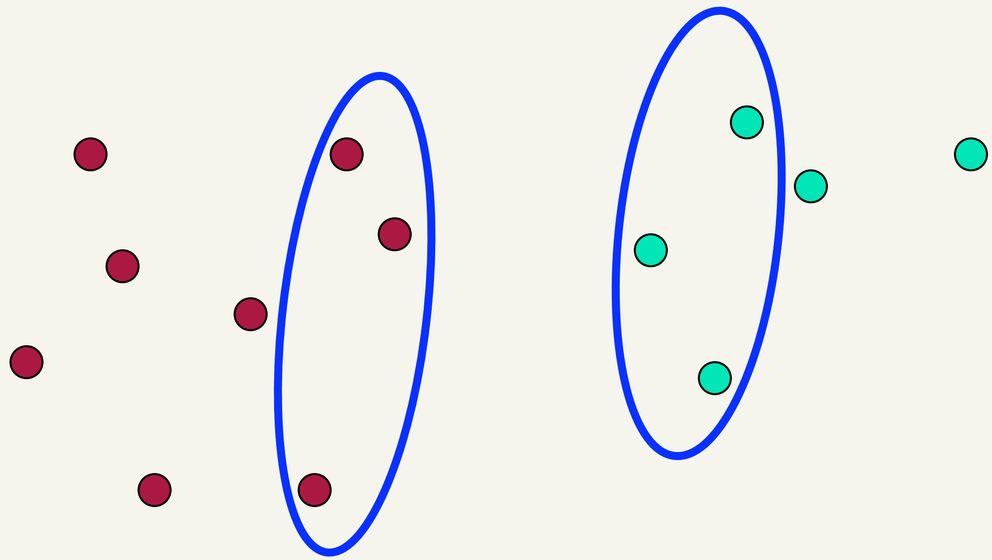


lots of possible solutions

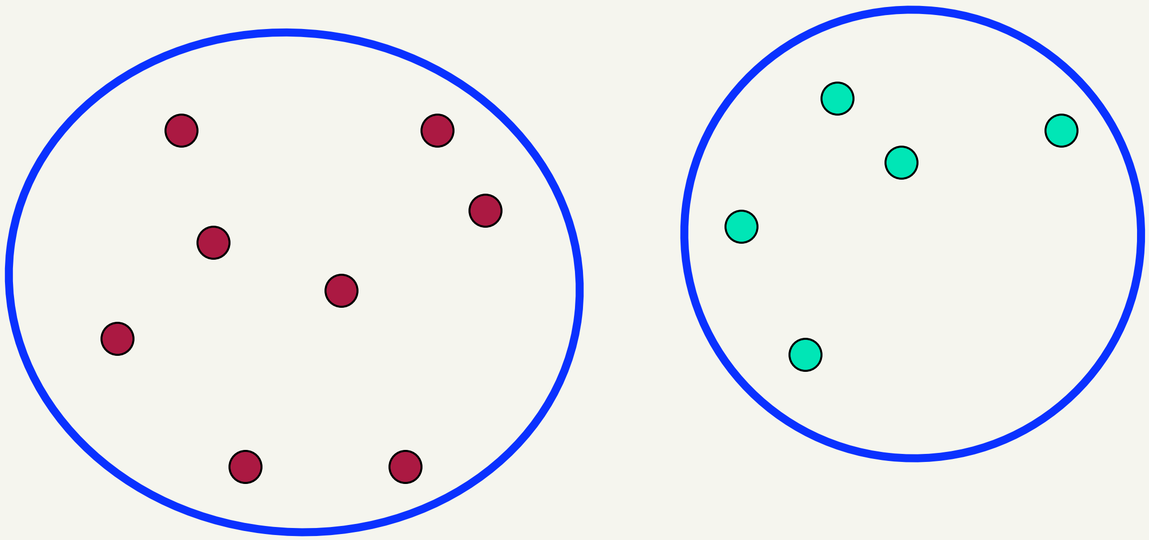
Which examples are important?



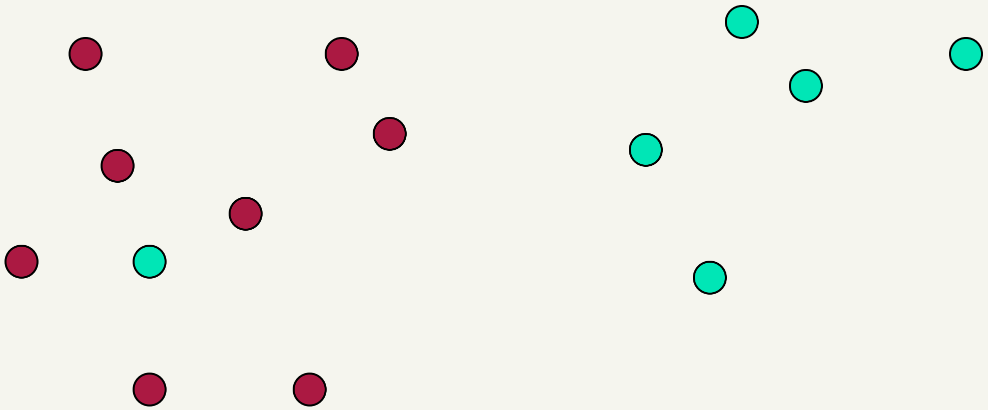
Which examples are important?



Which examples are important?

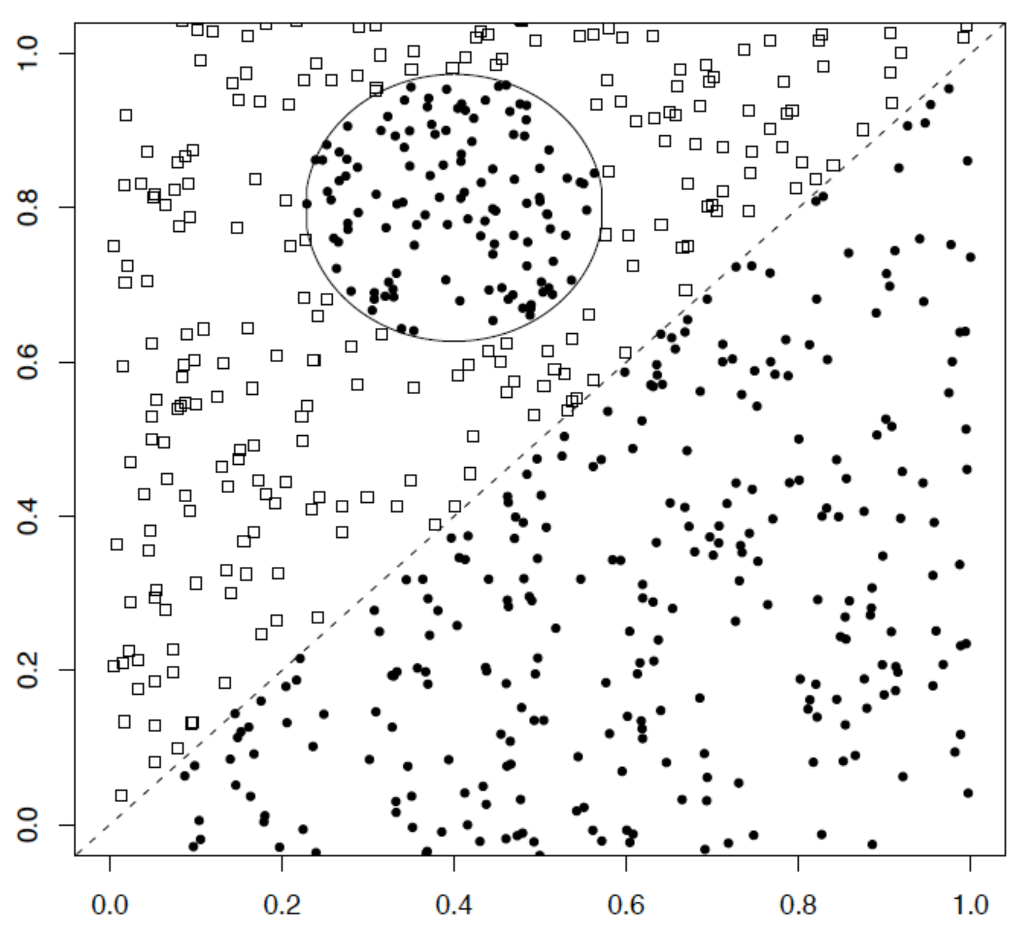


Dealing with noise



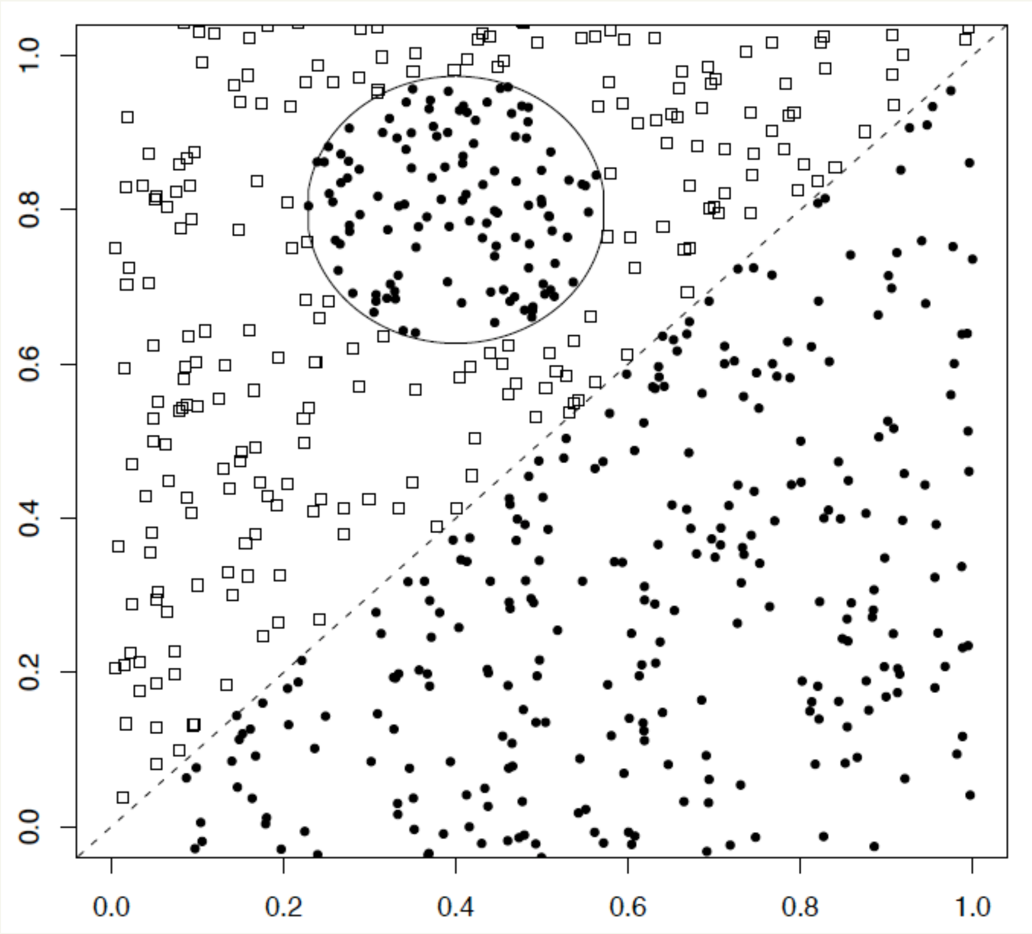
linearly separable?

A nonlinear problem



- A linear classifier like Naïve Bayes does badly on this task
- k-NN will do very well (assuming enough training data)

A nonlinear problem

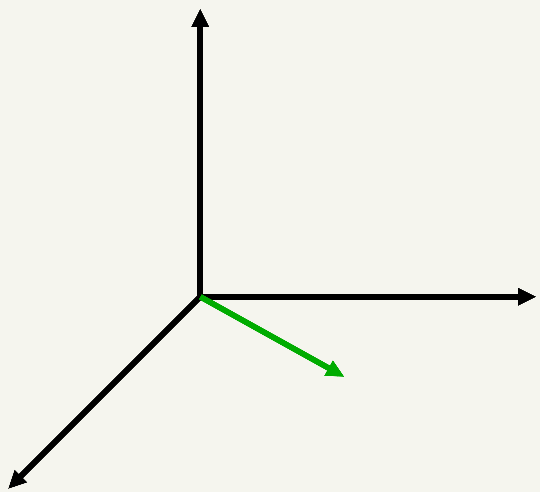


For text applications non-linear methods often do not perform better than linear

Why not?

High Dimensional Data

- Pictures like we’ve seen are misleading!
- Documents are zero along almost all axes
- Most document pairs are very far apart (i.e., not strictly orthogonal, but only share very common words and a few scattered others)
- In classification terms: often document sets are separable, for most any classification
- This is part of why linear classifiers are quite successful in this domain

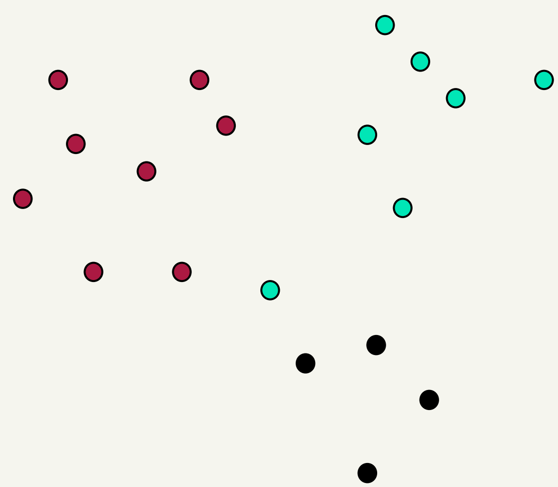


Dealing with multiple classes

- Scenarios
  - Document can belong to **zero or more classes**
  - Document must belong to exactly one class
- How can we do this?

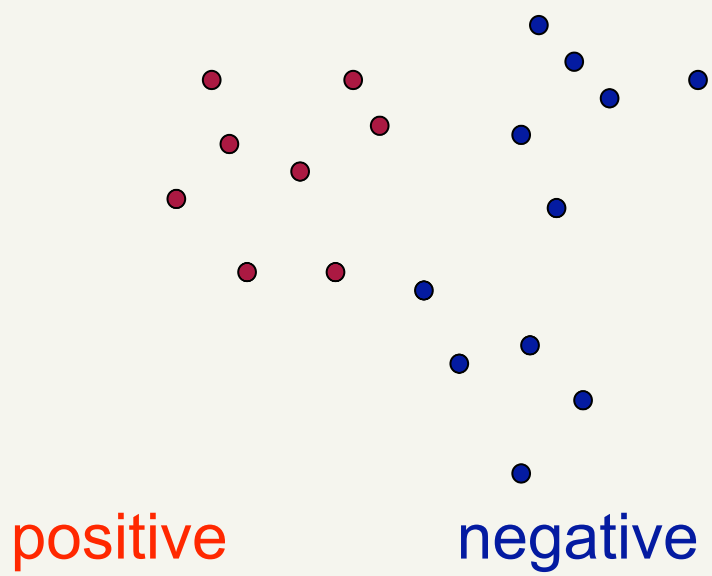
Set of Binary Classifiers

- Build a separator between each class and its complementary set (docs from all other classes).



Set of Binary Classifiers

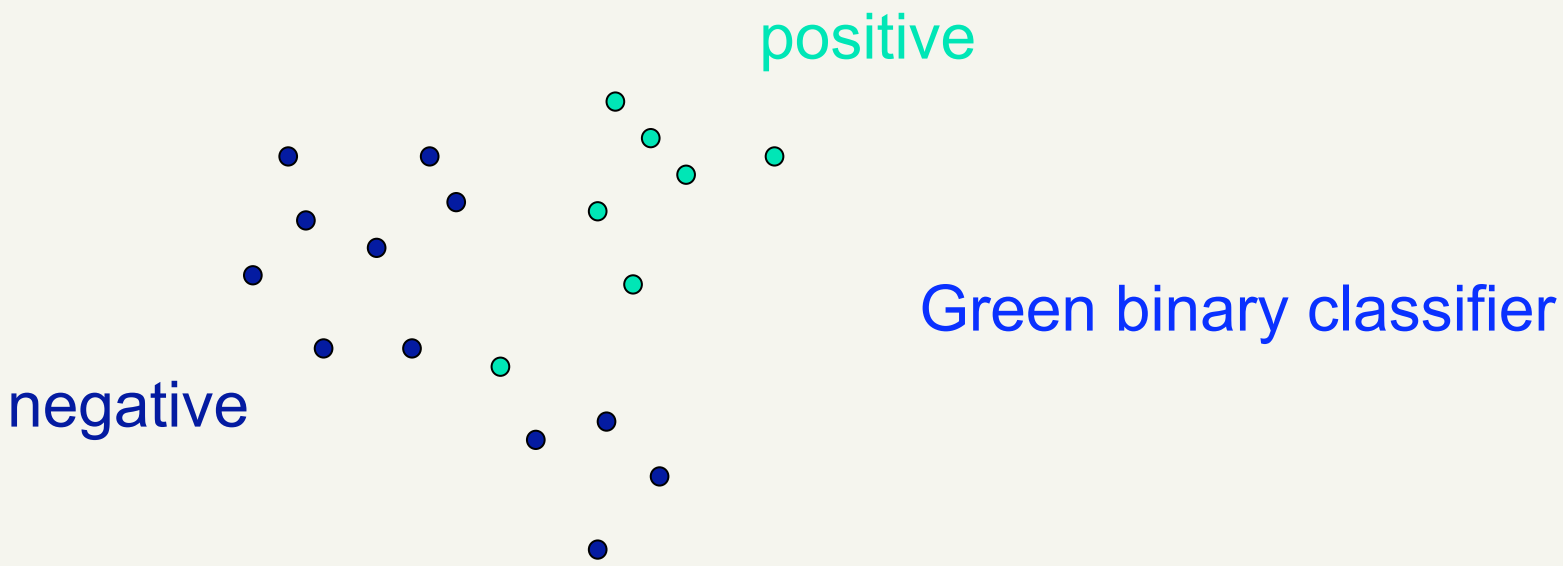
- Build a separator between each class and its complementary set (docs from all other classes).



Red binary classifier

Set of Binary Classifiers

- Build a separator between each class and its complementary set (docs from all other classes).



Set of Binary Classifiers

- Given a test doc, evaluate it for membership in each class with each binary classifier
- Assign document to class with:
  - maximum score
  - maximum confidence
  - maximum probability
  - threshold of the above
- Why different from multiclass/ any of classification?