CS276B

Text Information Retrieval, Mining, and Exploitation

Lecture 5

23 January 2003

Recap

Today’s topics

- Feature selection for text classification
- Measuring classification performance
- Nearest neighbor categorization

Feature Selection: Why?

- Text collections have a large number of features
  - 10,000 – 1,000,000 unique words – and more
- Make using a particular classifier feasible
  - Some classifiers can't deal with 100,000s of feat's
- Reduce training time
  - Training time for some methods is quadratic or worse in the number of features (e.g., logistic regression)
- Improve generalization
  - Eliminate noise features

Recap: Feature Reduction

- Standard ways of reducing feature space for text
  - Stemming
    - Laugh, laughs, laughing, laughed -> laugh
  - Stop word removal
    - E.g., eliminate all prepositions
  - Conversion to lower case
  - Tokenization
    - Break on all special characters: fire-fighter -> fire, fighter

Feature Selection

- Yang and Pedersen 1997
- Comparison of different selection criteria
  - DF – document frequency
  - IG – information gain
  - MI – mutual information
  - CHI – chi square
- Common strategy
  - Compute statistic for each term
  - Keep n terms with highest value of this statistic

Information Gain

$$\begin{aligned} G(t) = & -\sum_{i=1}^{m} P_r(c_i) \log P_r(c_i) \\ & + P_r(t) \sum_{i=1}^{m} P_r(c_i|t) \log P_r(c_i|t) \\ & + P_r(\bar{t}) \sum_{i=1}^{m} P_r(c_i|\bar{t}) \log P_r(c_i|\bar{t}) \end{aligned}$$

(Pointwise) Mutual Information

$$I(t,c) = \log \frac{P_r(t \wedge c)}{P_r(t) \times P_r(c)}$$

$$I_{avg}(t) = \sum_{i=1}^{m} P_r(c_i) I(t, c_i)$$

$$I_{max}(t) = \max_{i=1}^{m} \{ I(t, c_i) \}$$

Chi-Square

	Term present	Term absent
Document belongs to category	A	B
Document does not belong to category	C	D

$X^2 = N(AD-BC)^2 / ((A+B) (A+C) (B+D) (C+D))$

Use either maximum or average X^2

Value for complete independence?

Document Frequency

- Number of documents a term occurs in
- Is sometimes used for eliminating both very frequent and very infrequent terms
- How is document frequency measure different from the other 3 measures?

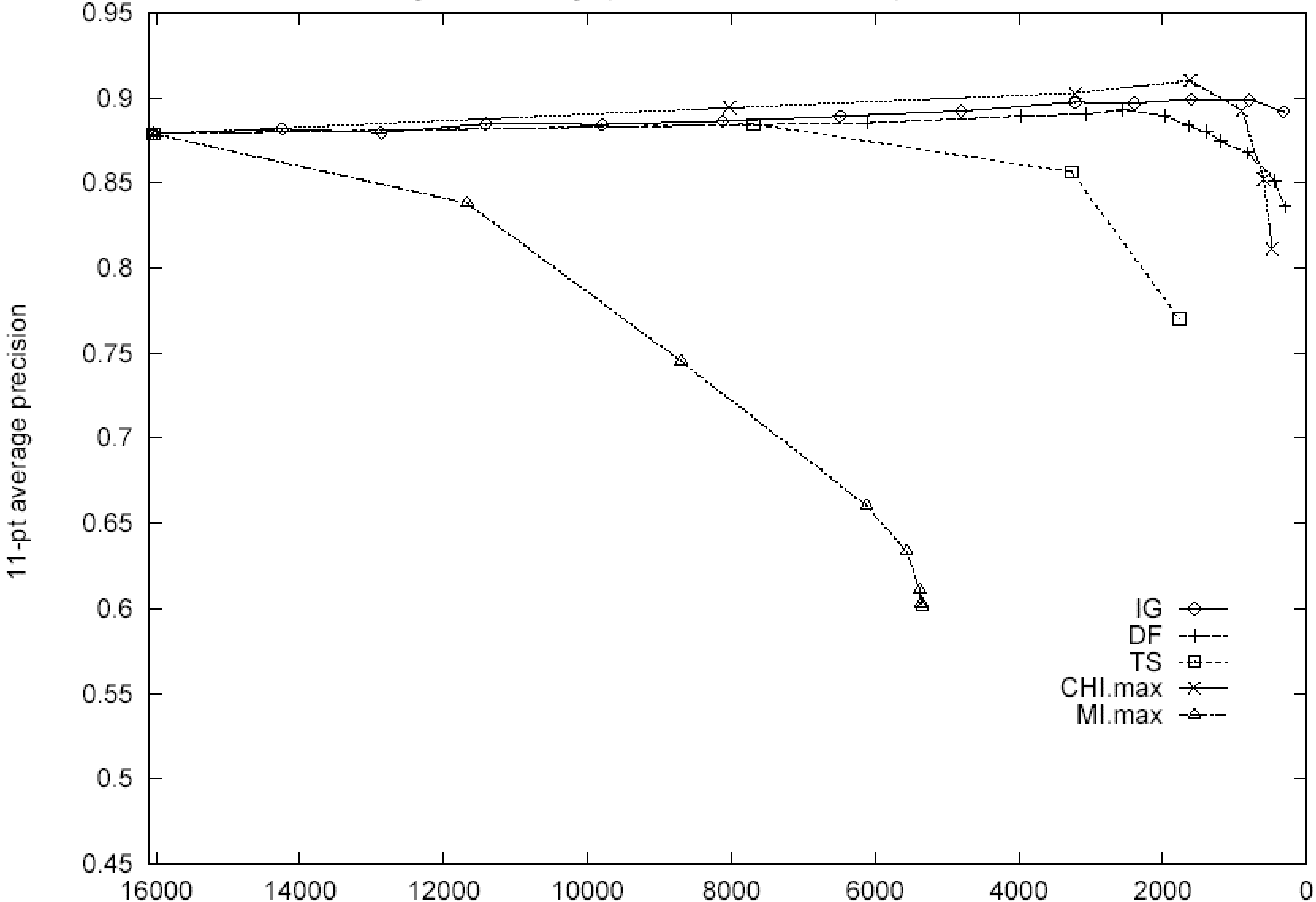
Yang&Pedersen: Experiments

- Two classification methods
  - kNN (k nearest neighbors; more later)
  - Linear Least Squares Fit
    - Regression method
- Collections
  - Reuters-22173
    - 92 categories
    - 16,000 unique terms
  - Ohsumed: subset of medline
    - 14,000 categories
    - 72,000 unique terms
- Ltc term weighting

Yang&Pedersen: Experiments

- Choose feature set size
- Preprocess collection, discarding non-selected features / words
- Apply term weighting -> feature vector for each document
- Train classifier on training set
- Evaluate classifier on test set

Figure 1. Average precision of kNN vs. unique word count



Discussion

- You can eliminate 90% of features for IG, DF, and CHI without decreasing performance.
- In fact, performance **increases** with fewer features for IG, DF, and CHI.
- Mutual information is very sensitive to small counts.
- IG does best with smallest number of features.
- Document frequency is close to optimal. By far the simplest feature selection method.
- Similar results for LLSF (regression).

Results

Table 1. Criteria and performance of feature selection methods in kNN & LLSF

Method	DF	IG	CHI	MI	TS
favoring common terms	Y	Y	Y	N	Y/N
using categories	N	Y	Y	Y	N
using term absence	N	Y	Y	N	N
performance in kNN/LLSF	excellent	excellent	excellent	poor	ok

Why is selecting common terms a good strategy?

IG, DF, CHI Are Correlated.

Figure 3. Correlation between DF and IG values of words in Reuters

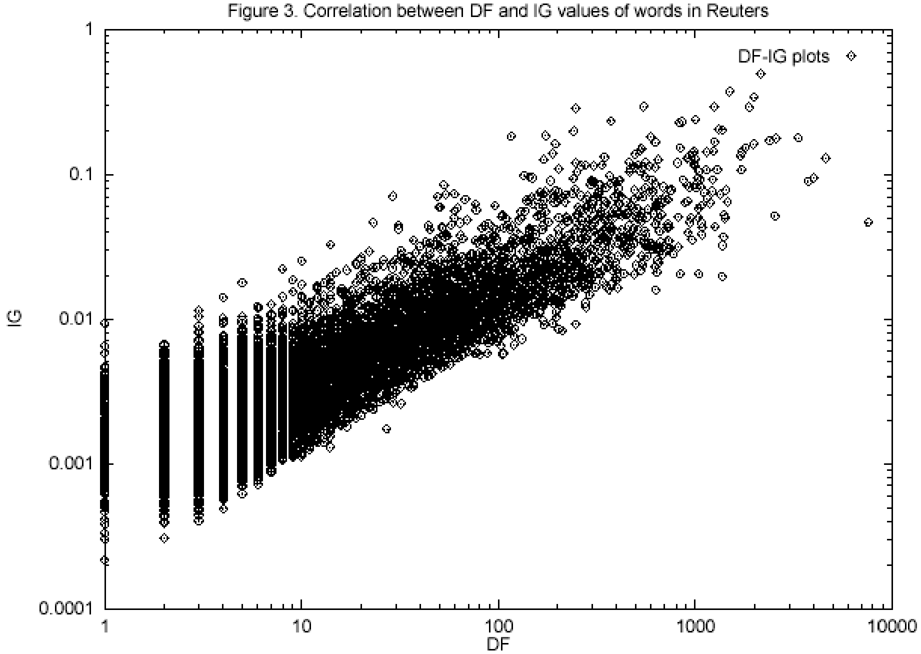
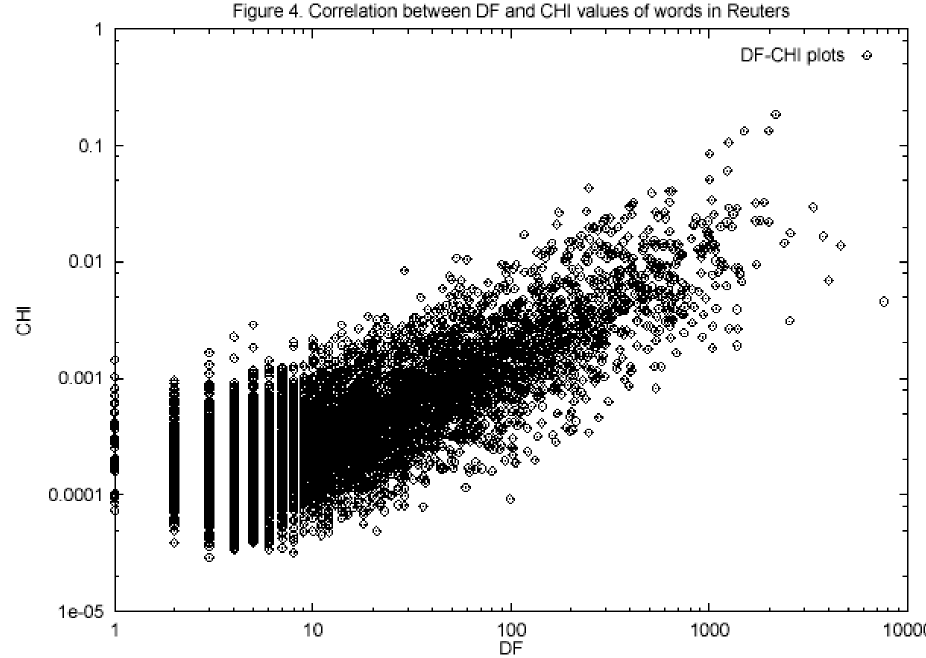


Figure 4. Correlation between DF and CHI values of words in Reuters



Information Gain vs Mutual Information

- Information gain is similar to MI for random variables
- Independence?
- In contrast, pointwise MI ignores non-occurrence of terms
  - E.g., for complete dependence, you get:
  - P(AB)/ (P(A)P(B)) = 1/P(A) – larger for rare terms than for frequent terms
- Yang&Pedersen: Pointwise MI favors rare terms

$$G(t) = \sum_{X \in \{t, \bar{t}\}} \sum_{Y \in \{c_i\}} P_r(X, Y) \log \frac{P_r(X, Y)}{P_r(X) P_r(Y)}$$

Feature Selection: Other Considerations

- Generic vs Class-Specific
  - Completely generic (class-independent)
  - Separate feature set for each class
  - Mixed (a la Yang&Pedersen)
- Maintainability over time
  - Is aggressive features selection good or bad for robustness over time?
- Ideal: Optimal features selected as part of training

Yang&Pedersen: Limitations

- Don’t look at class specific feature selection
- Don’t look at methods that can’t handle high-dimensional spaces
- Evaluate category ranking (as opposed to classification accuracy)

Feature Selection: Other Methods

- Stepwise term selection
  - Forward
  - Backward
  - Expensive: need to do n^2 iterations of training
- Term clustering
- Dimension reduction: PCA / SVD

Word Rep. vs. Dimension Reduction

- Word representations: one dimension for each word (binary, count, or weight)
- Dimension reduction: each dimension is a unique linear combination of all words (linear case)
- Dimension reduction is good for generic topics (“politics”), bad for specific classes (“ruanda”). Why?
- SVD/PCA computationally expensive
- Higher complexity in implementation
- No clear examples of higher performance through dimension reduction

Word Rep. vs. Dimension Reduction

classifier	input	precision for Topics 51–100 average	% change	at 100	precision for Topics 101–150 average	% change	at 100	average change
baseline	expansion	0.3678	+0.0%	0.4710	0.3705	+0.0%	0.4194	+0
	LSI	0.3240	-11.9%	0.4210	0.3268	-11.8%	0.3908	-12
	200 terms	0.3789	+3.0%	0.4824	0.3712	+0.2%	0.4440	+2
	LSI + 200 terms	0.3359	-8.7%	0.4426	0.3358	-9.4%	0.3928	-9
logistic regression	LSI	0.3980	+8.2%	0.5108	0.4057	+9.5%	0.4802	+9
	200 terms	0.3654	-0.7%	0.4788	0.3637	-1.8%	0.4434	-1
	LSI + 200 terms	0.3494	-5.0%	0.4652	0.3457	-6.7%	0.4168	-6
LDA	LSI	0.4139	+12.5%	0.5166	0.4230	+14.2%	0.4870	+13
	200 terms	0.3966	+7.8%	0.4916	0.3841	+3.7%	0.4586	+6
	LSI + 200 terms	0.3973	+8.0%	0.5034	0.3910	+5.5%	0.4616	+7
linear network	LSI	0.4098	+11.4%	0.5094	0.4211	+13.7%	0.4830	+13
	200 terms	0.4209	+14.4%	0.5044	0.4121	+11.2%	0.4742	+13
	LSI + 200 terms	0.4273	+16.2%	0.5180	0.4302	+16.1%	0.4908	+16
non-linear network	LSI	0.4110	+11.7%	0.5090	0.4208	+13.6%	0.4834	+13
	200 terms	0.4210	+14.5%	0.5026	0.4115	+11.1%	0.4740	+13
	LSI + 200 terms	0.4251	+15.6%	0.5204	0.4318	+16.5%	0.4882	+16

Measuring Classification Figures of Merit

- Accuracy of classification
  - Main evaluation criterion in academia
  - More in a momen
- Speed of training statistical classifier
- Speed of classification (docs/hour)
  - No big differences for most algorithms
  - Exceptions: kNN, complex preprocessing requirements
- Effort in creating training set (human hours/topic)
  - More on this in Lecture 9 (Active Learning)

Measures of Accuracy

- Error rate
  - Not a good measure for small classes. Why?
- Precision/recall for classification decisions
- F_1 measure: $1/F_1 = ½ (1/P + 1/R)$
- Breakeven point
- Correct estimate of size of category
  - Why is this different?
- Precision/recall for ranking classes
- Stability over time / concept drift
- Utility

Precision/Recall for Ranking Classes

- Example: “Bad wheat harvest in Turkey”
- True categories
  - Wheat
  - Turkey
- Ranked category list
  - 0.9: turkey
  - 0.7: poultry
  - 0.5: armenia
  - 0.4: barley
  - 0.3: georgia
- Precision at 5: 0.1, Recall at 5: 0.5

Precision/Recall for Ranking Classes

- Consider problems with many categories (>10)
- Use method returning scores comparable across categories (not: Naïve Bayes)
- Rank categories and compute average precision recall (or other measure characterizing precision/recall curve)
- Good measure for interactive support of human categorization
- Useless for an “autonomous” system (e.g. a filter on a stream of newswire stories)

Concept Drift

- Categories change over time
- Example: “president of the united states”
  - 1999: clinton is great feature
  - 2002: clinton is bad feature
- One measure of a text classification system is how well it protects against concept drift.
- Feature selection: good or bad to protect against concept drift?

Micro- vs. Macro-Averaging

- If we have more than one class, how do we combine multiple performance measures into one quantity?
- Macroaveraging: Compute performance for each class, then average.
- Microaveraging: Collect decisions for all classes, compute contingency table, evaluate.

Micro- vs. Macro-Averaging: Example

Class 1

	Truth: yes	Truth: no
Classifier: yes	10	10
Classifier: no	10	970

Class 2

	Truth: yes	Truth: no
Classifier: yes	90	10
Classifier: no	10	890

Micro.Av. Table

	Truth: yes	Truth: no
Classifier: yes	100	20
Classifier: no	20	1860

- Macroaveraged precision: (0.5 + 0.9)/2 = 0.7
- Microaveraged precision: 100/120 = .83
- Why this difference?

Reuters 1

- Newswire text
- Statistics (vary according to version used)
  - Training set: 9,610
  - Test set: 3,662
  - 50% of documents have no category assigned
  - Average document length: 90.6
  - Number of classes: 92
  - Example classes: currency exchange, wheat, gold
  - Max classes assigned: 14
  - Average number of classes assigned
    - 1.24 for docs with at least one category

Reuters 1

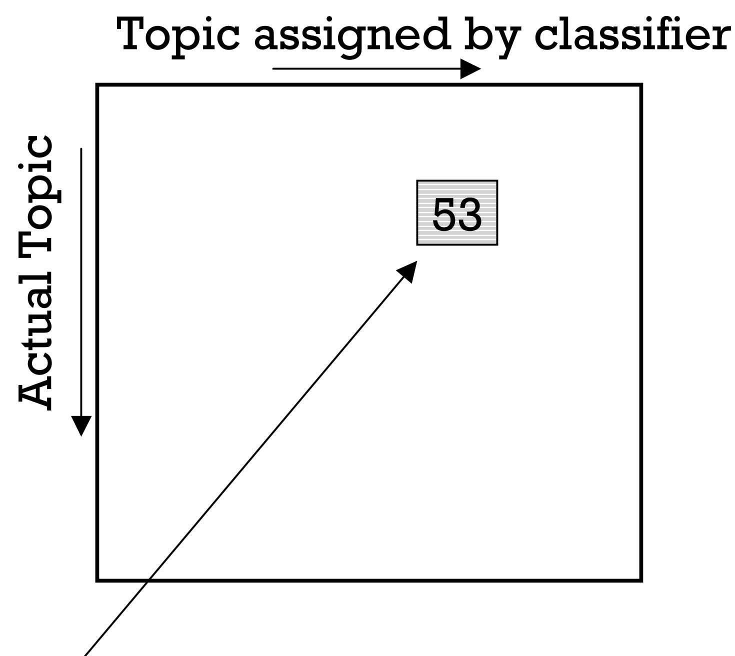
- Only about 10 out of 92 categories are large
- Microaveraging measures performance on large categories.

Factors Affecting Measures

- Variability of data
  - Document size/length
  - quality/style of authorship
  - uniformity of vocabulary
- Variability of “truth” / gold standard
  - need definitive judgement on which topic(s) a doc belongs to
    - usually human
  - Ideally: consistent judgements

Accuracy measurement

- Confusion matrix



This (i, j) entry means 53 of the docs actually in topic i were put in topic j by the classifier.

Confusion matrix

- Function of classifier, topics and test docs.
- For a perfect classifier, all off-diagonal entries should be zero.
- For a perfect classifier, if there are n docs in category j than entry (j,j) should be n.
- Straightforward when there is 1 category per document.
- Can be extended to n categories per document.

Confusion measures (1 class / doc)

- Recall: Fraction of docs in topic *i* classified correctly: $$\frac{c_{ii}}{\sum_j c_{ij}}$$

- Precision: Fraction of docs assigned topic *i* that are actually about topic *i*: $$\frac{c_{ii}}{\sum_j c_{ji}}$$

- "Correct rate": (1- error rate) Fraction of docs classified correctly: $$\frac{\sum_i c_{ii}}{\sum_j \sum_i c_{ij}}$$

Integrated Evaluation/Optimization

$$\mathrm{E}[h(\vec{s},\vec{Z})] = \sum_{\vec{z}\in\{0,1\}^n} P(\vec{Z}=\vec{z})h(\vec{s},\vec{z})$$

$$= \sum_{\vec{z}\in\{0,1\}^n} (\prod_{i=1}^{n} p_i^{z_i}(1-p_i)^{(1-z_i)})h(\vec{s},\vec{z})$$

- Principled approach to training
  - Optimize the measure that performance is measured with
- s: vector of classifier decision, z: vector of true classes
- h(s,z) = cost of making decisions s for true

Utility / Cost

- One cost function h is based on contingency table.
- Assume identical cost for all false positives etc.
- Cost C = l11 * A + l12 *B + l21*C + l22*D
- For this cost c, we get the following optimality criterion

$$p_i > \frac{(\lambda_{12} - \lambda_{22})}{(\lambda_{21} - \lambda_{11}) + (\lambda_{12} - \lambda_{22})}.$$

	Truth: yes	Truth: no
Classifier: yes	Cost:λ11 Count:A	Cost:λ12 Count:B
Classifier: no	Cost:λ21 Count;C	Cost:λ22 Count:D

Utility / Cost

$$p_i > \frac{(\lambda_{12} - \lambda_{22})}{(\lambda_{21} - \lambda_{11}) + (\lambda_{12} - \lambda_{22})}.$$

Most common cost: 1 for error, 0 for correct. Pi > ?

Product cross-sale: high cost for false positive, low cost for false negative.

Patent search: low cost for false positive, high cost for false negative.

	Truth: yes	Truth: no
Classifier: yes	λ_{11}	λ_{12}
Classifier: no	λ_{21}	λ_{22}

Are All Optimal Rules of Form $p>\theta$?

- In the above examples, all you need to do is estimate probability of class membership.
- Can all problems be solved like this?
- No!
- Probability is often not sufficient
- User decision depends on the distribution of relevance
- Example: information filter for terrorism

Naïve Bayes

Vector Space Classification
Nearest Neighbor Classification

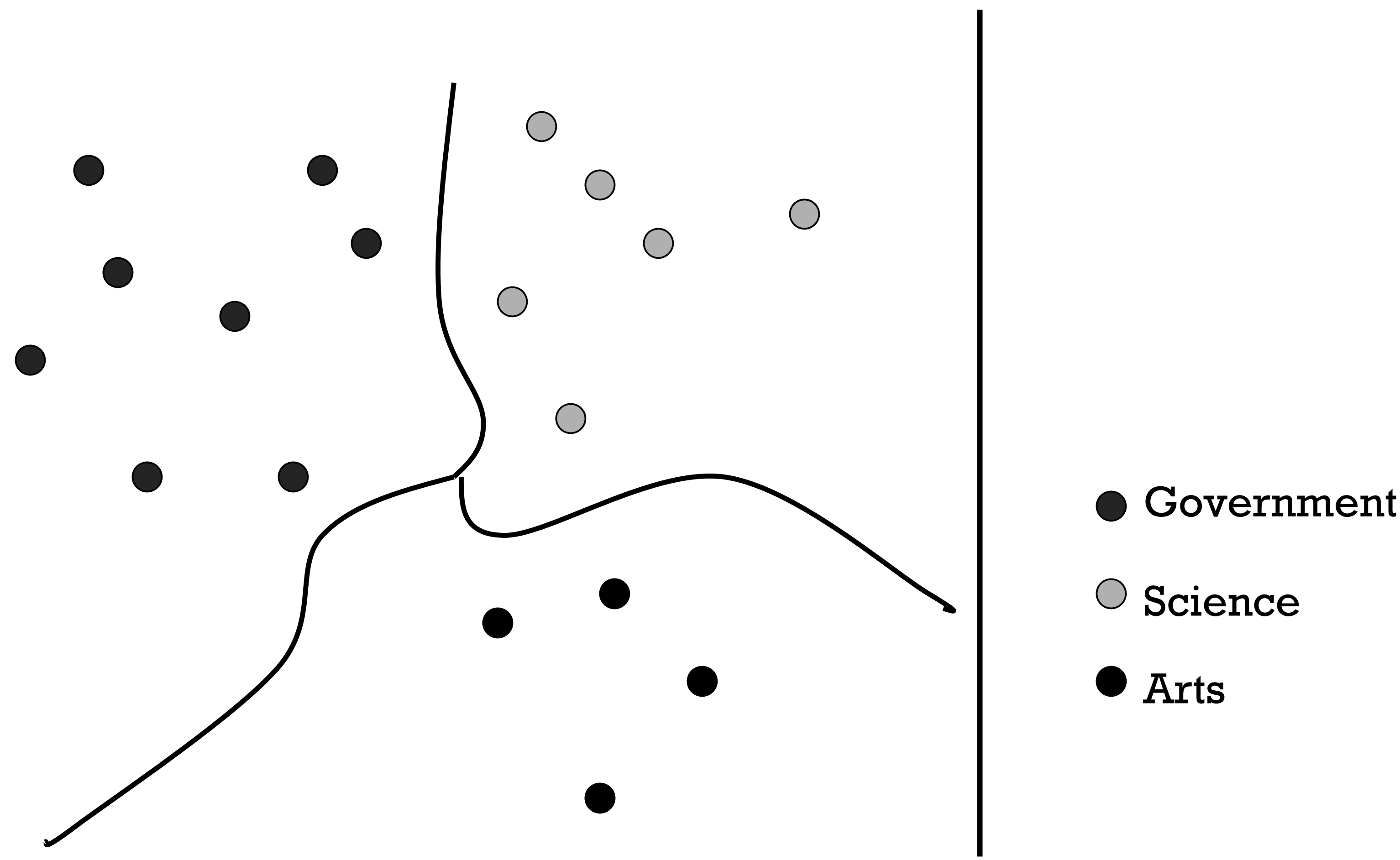
Recall Vector Space Representation

- Each doc j is a vector, one component for each term (= word).
- Normalize to unit length.
- Have a vector space
  - terms are axes
  - n docs live in this space
  - even with stemming, may have 10000+ dimensions, or even 1,000,000+

Classification Using Vector Spaces

- Each training doc a point (vector) labeled by its topic (= class)
- Hypothesis: docs of the same topic form a contiguous region of space
- Define surfaces to delineate topics in space

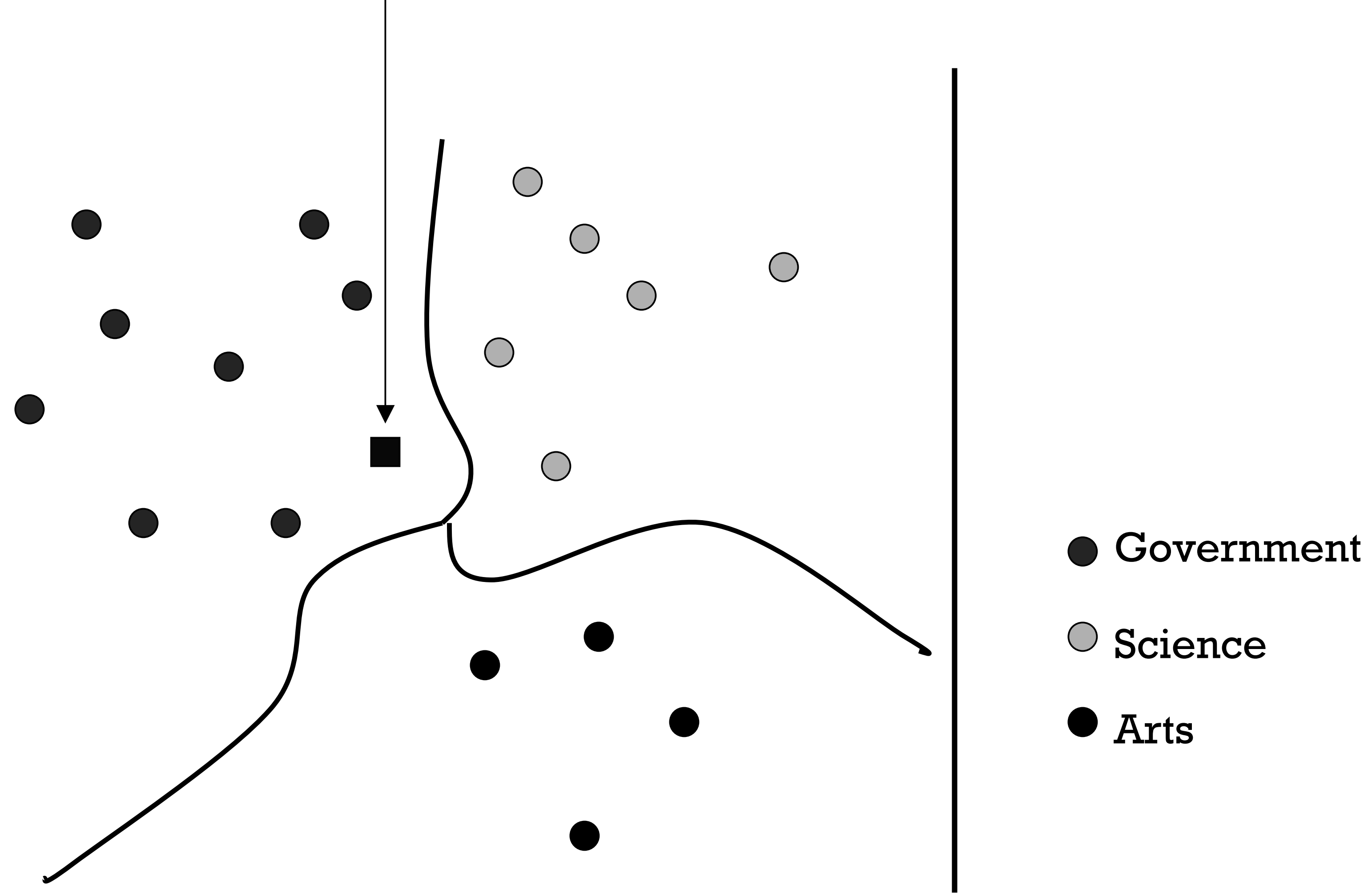
Topics in a vector space



Given a test doc

- Figure out which region it lies in
- Assign corresponding class

Test doc = Government



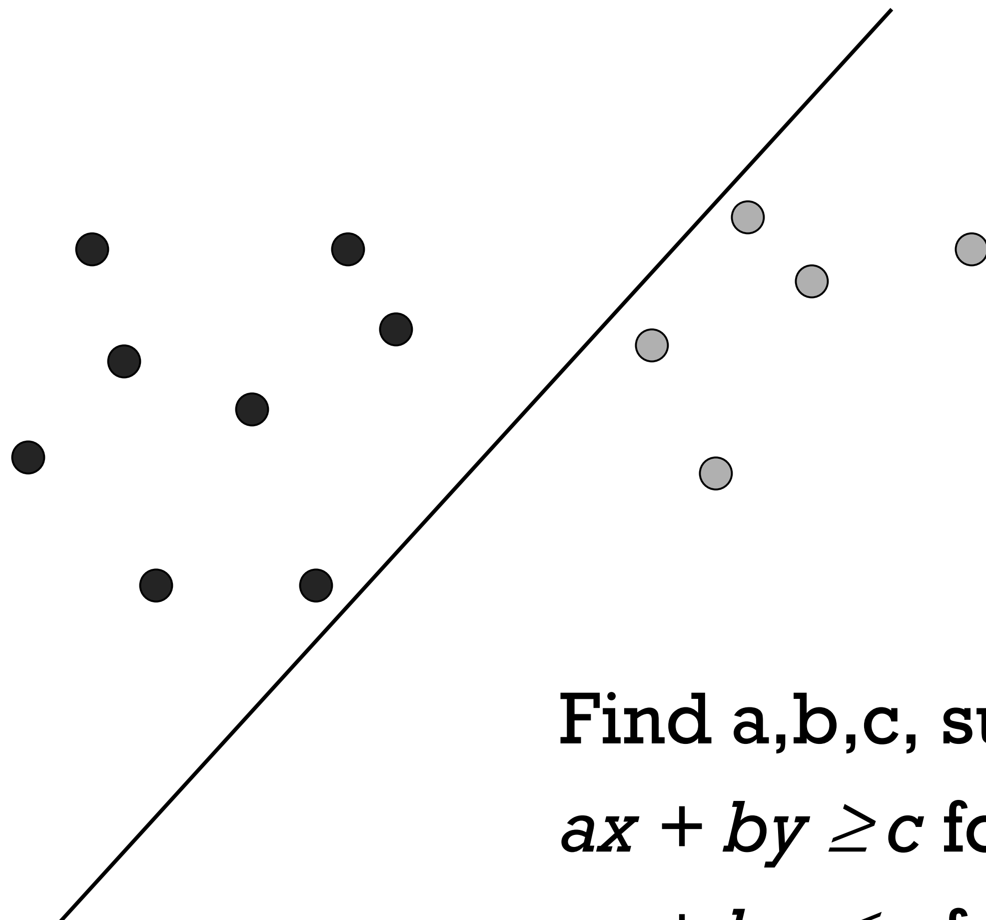
Binary Classification

- Consider 2 class problems
- How do we define (and find) the separating surface?
- How do we test which region a test doc is in?

Separation by Hyperplanes

- Assume *linear separability* for now:
  - in 2 dimensions, can separate by a line
  - in higher dimensions, need hyperplanes
- Can find separating hyperplane by *linear programming* (e.g. perceptron):
  - separator can be expressed as $ax + by = c$

Linear programming / Perceptron

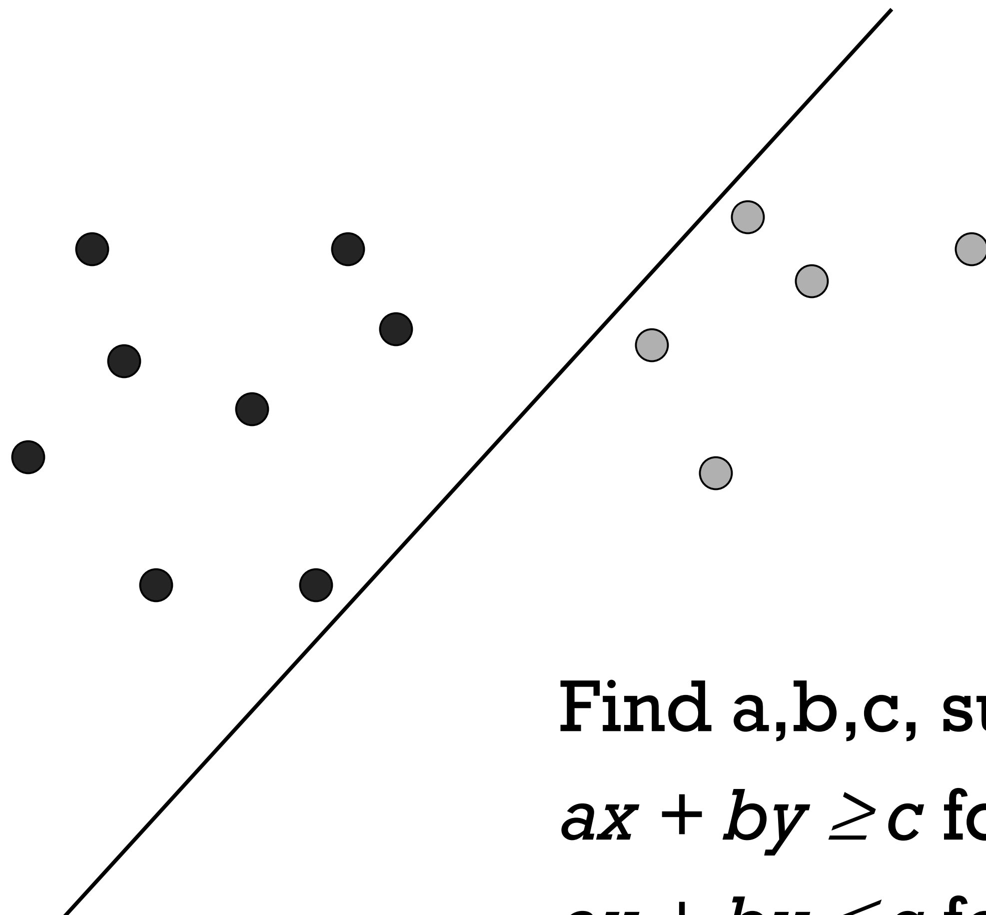


Find a,b,c, such that

$ax + by \geq c$ for red points

$ax + by \leq c$ for green points.

Relationship to Naïve Bayes?



Find a,b,c, such that

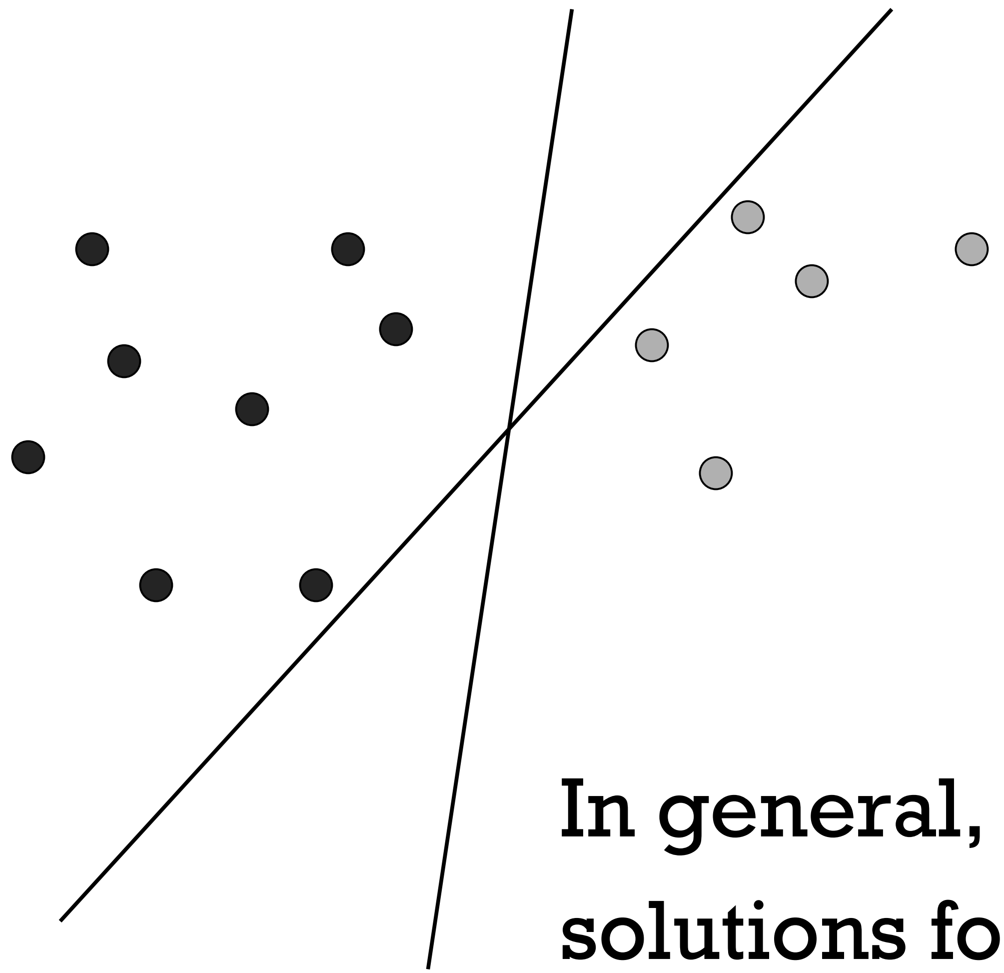
$ax + by \geq c$ for red points

$ax + by \leq c$ for green points.

Linear Classifiers

- Many common text classifiers are linear classifiers
- Despite this similarity, large performance differences
  - For separable problems, there is an infinite number of separating hyperplanes. Which one do you choose?
  - What to do for non-separable problems?

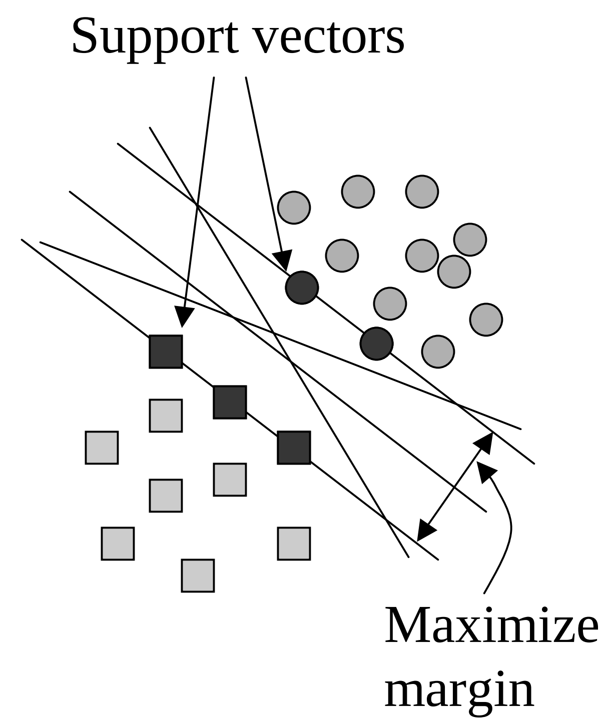
Which hyperplane?



In general, lots of possibl solutions for *a,b,c.*

Support Vector Machine (SVM)

- *Quadratic programming* problem
- The decision function is fully specified by subset of training samples, *the support vectors*.
- Text classification method du jour
- Topic of lecture 9



Category: Interest

- **Example SVM features**
- **w_i t_i** **w_i t_i**

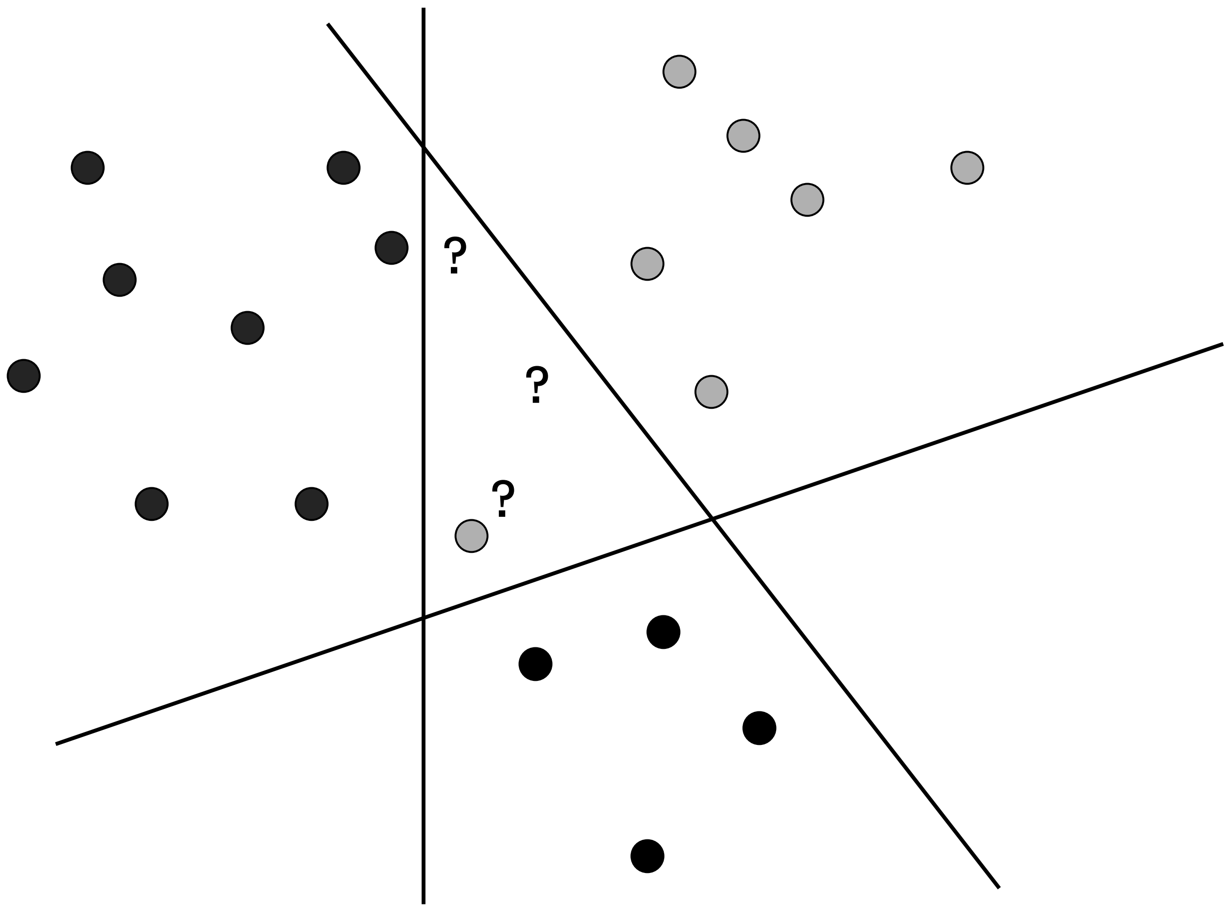
- 0.70 prime
- 0.67 rate
- 0.63 interest
- 0.60 rates
- 0.46 discount
- 0.43 bundesbank
- 0.43 baker

- -0.71 dlrs
- -0.35 world
- -0.33 sees
- -0.25 year
- -0.24 group
- -0.24 dlr
- -0.24 january

More Than Two Classes

- Any-of or multiclass classification
  - For n classes, decompose into n binary problems
- One-of classification: each document belongs to exactly one class
  - How do we compose separating surfaces into regions?
- Centroid classification
- K nearest neighbor classification

Composing Surfaces: Issues

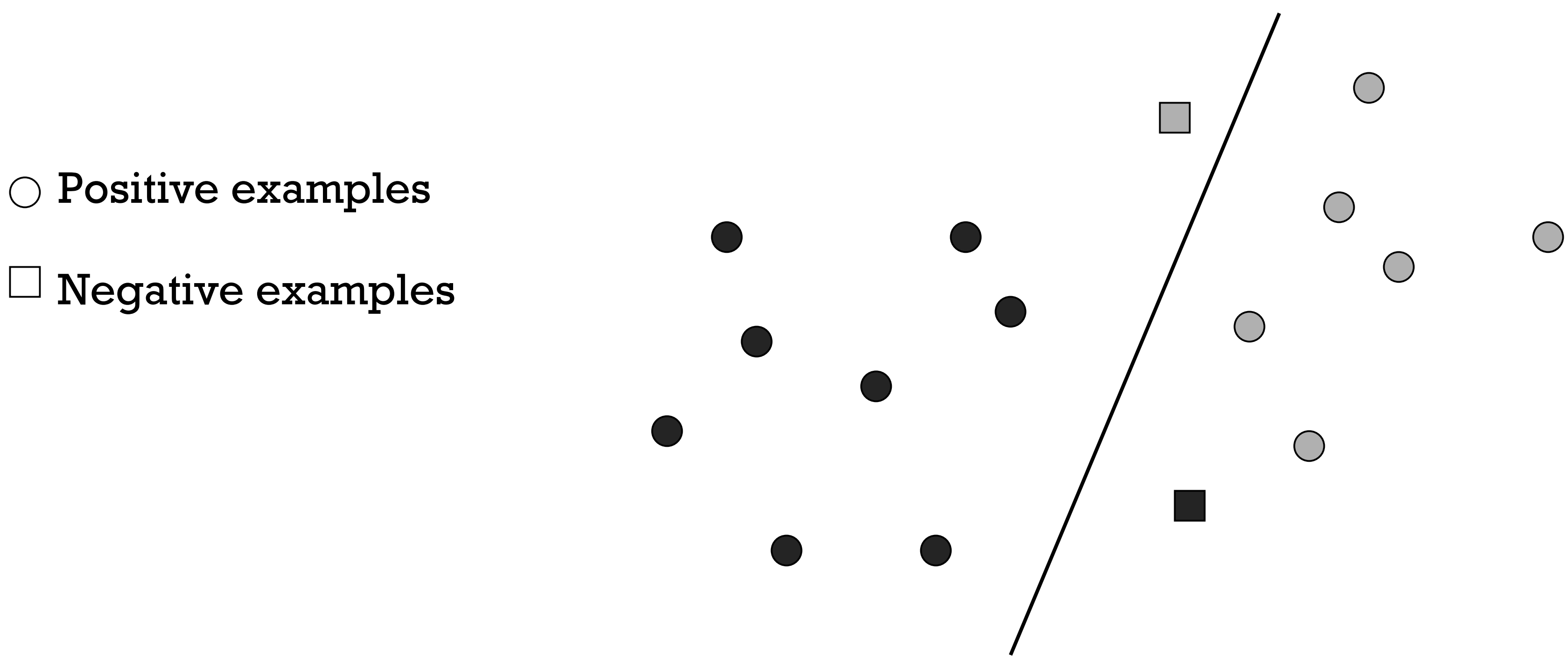


Separating Multiple Topics

- Build a separator between each topic and its complementary set (docs from all other topics).
- Given test doc, evaluate it for membership in each topic.
- Declare membership in topics
  - One-of classification:
    - for class with maximum score/confidence/probability
  - Multiclass classification:
    - For classes above threshold

Negative examples

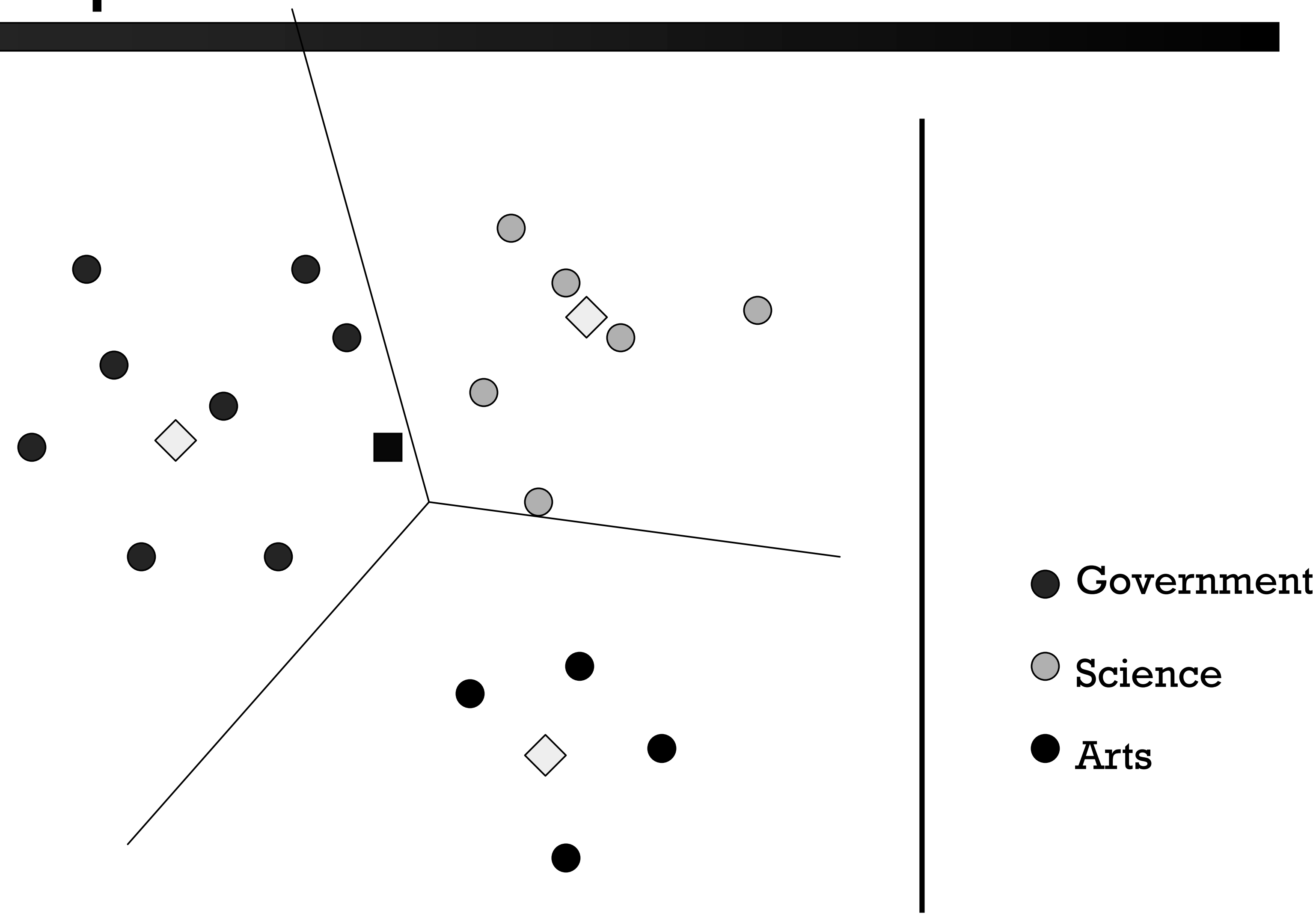
- Formulate as above, except negative examples for a topic are added to its complementary set.



Centroid Classification

- Given training docs for a topic, compute their centroid
- Now have a centroid for each topic
- Given query doc, assign to topic whose centroid is nearest.
- Exercise: Compare to Rocchio

Example



k Nearest Neighbor Classification

- To classify document d into class c
- Define k-neighborhood N as k nearest neighbors of d
- Count number of documents l in N that belong to c
- Estimate P(c|d) as l/k

Cover and Hart 1967

- Asymptotically, the error rate of 1-nearest-neighbor classification is less than twice the Bayes rate.
- Assume that query point coincides with a training point.
- Both query point and training point contribute error -> 2 times Bayes rate

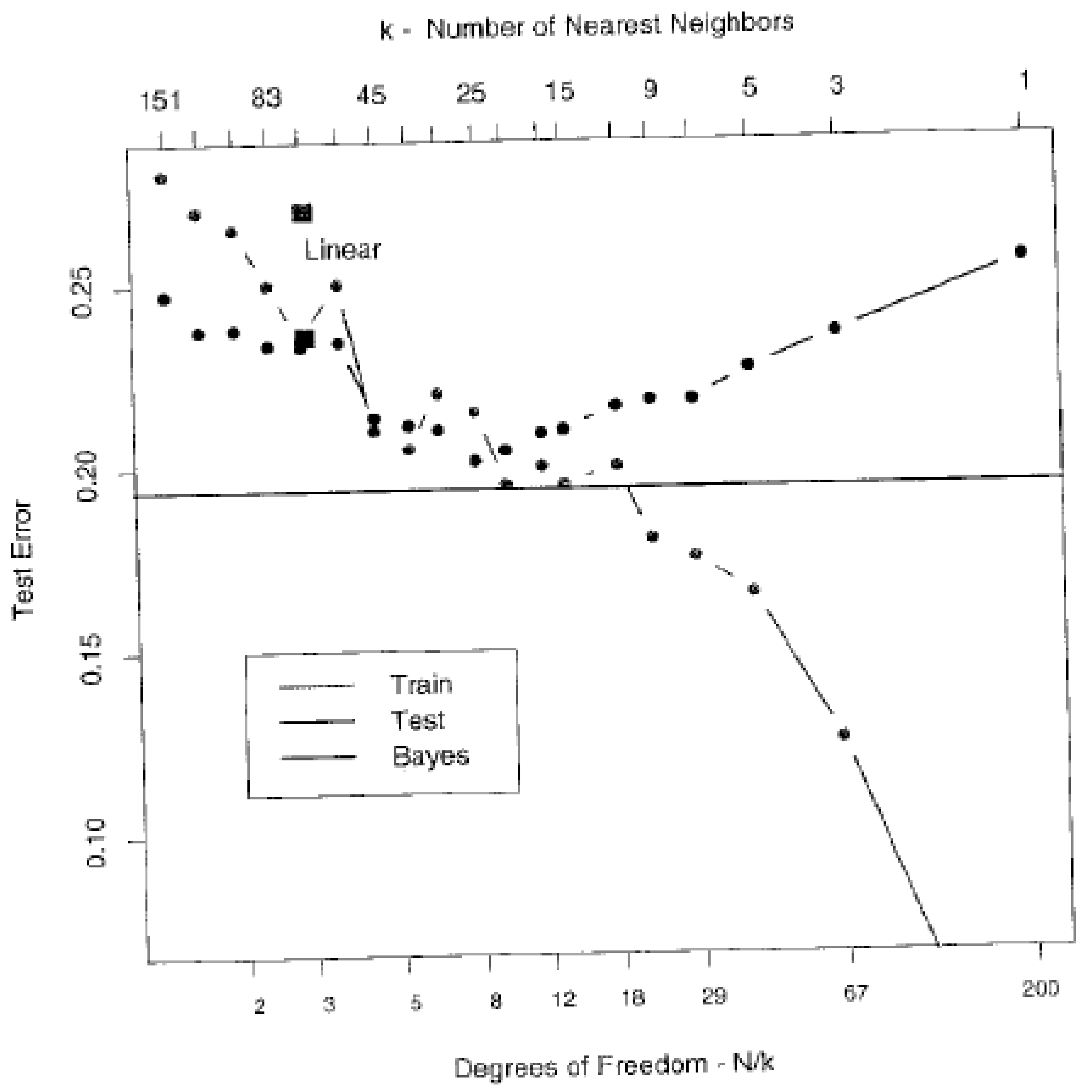
kNN vs. Regression

- kNN has high variance and low bias.
- Linear regression has low variance and high bias.

kNN: Discussion

- Classification time linear in training set
- Training set generation
  - incompletely judged set can be problematic for multiclass problems
- No feature selection necessary
- Scales well with large number of categories
  - Don’t need to train n classifiers for n classes
- Categories can influence each other
  - Small changes to one category can have ripple effect
- Scores can be hard to convert to probabilities
- No training necessary
  - Actually: not true. Why?

Number of neighbors



References

- A Comparative Study on Feature Selection in Text Categorization (1997) Yiming Yang, Jan O. Pedersen. Proceedings of ICML-97, 14th International Conference on Machine Learning.
- Evaluating and Optimizing Autonomous Text Classification Systems (1995) David Lewis. Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval
- Foundations of Statistical Natural Language Processing. Chapter 16. MIT Press. Manning and Schuetze.
- Trevor Hastie, Robert Tibshirani and Jerome Friedman, "Elements of Statistical Learning: Data Mining, Inference and Prediction" Springer Verlag

Kappa Measure

- Kappa measures
  - Agreement among coders
  - Designed for categorical judgments
  - Corrects for chance agreement
- Kappa = [P(A) – P(E)] / [1 – P(E)]
- P(A) – proportion of time coders agree
- P(E) – what agreement would be by chance
- Kappa = 0 for chance agreement, 1 for total agreement.