

# Introduction to Information Retrieval

## Lecture 12: Support vector machines and machine learning on documents

[Borrows slides from Ray Mooney]

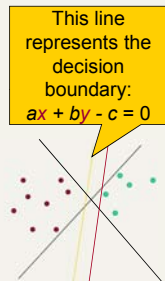
## Text classification: Last time and today

- Last time: 3 algorithms for text classification
  - K Nearest Neighbor classification
    - Simple, expensive at test time, high variance, non-linear
  - Vector space classification using centroids and hyperplanes that split them
    - Simple, linear classifier; perhaps too simple (or maybe not\*)
  - (Decision Trees)
    - Pick out hyperboxes; nonlinear; use just a few features
- Today
  - SVMs
  - Some empirical evaluation and comparison
  - Text-specific issues in classification

2

## Linear classifiers: Which Hyperplane?

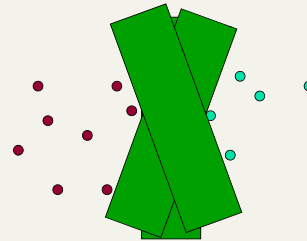
- Lots of possible solutions for  $a, b, c$ .
- Some methods find a separating hyperplane, but not the optimal one [according to some criterion of expected goodness]
  - E.g., perceptron
- Support Vector Machine (SVM) finds an optimal solution.
  - Maximizes the distance between the hyperplane and the "difficult points" close to decision boundary
  - One intuition: if there are no points near the decision surface, then there are no very uncertain classification decisions



15.0

## Another intuition

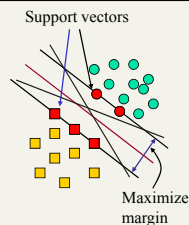
- If you have to place a fat separator between classes, you have less choices, and so the capacity of the model has been decreased



4

## Support Vector Machine (SVM)

- SVMs maximize the *margin* around the separating hyperplane.
  - A.k.a. large margin classifiers
- The decision function is fully specified by a subset of training samples, *the support vectors*.
- Solving SVMs is a *quadratic programming* problem
- Seen by many as the most successful current text classification method\*



\*but other discriminative methods often perform very similarly

15.1

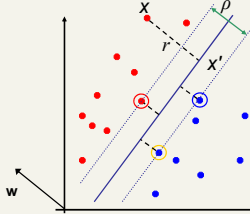
## Maximum Margin: Formalization

- $w$ : decision hyperplane normal vector
- $x_i$ : data point  $i$
- $y_i$ : class of data point  $i$  (+1 or -1) **NB: Not 1/0**
- Classifier is:  $f(x_i) = \text{sign}(w^T x_i + b)$
- Functional margin of  $x_i$  is:  $y_i (w^T x_i + b)$ 
  - But note that we can increase this margin simply by scaling  $w, b, \dots$
- Functional margin of dataset is twice the minimum functional margin for any point
  - The factor of 2 comes from measuring the whole width of the margin

6

## Geometric Margin

- Distance from example to the separator is  $r = y \frac{\mathbf{w}^T \mathbf{x} + b}{\|\mathbf{w}\|}$
- Examples closest to the hyperplane are **support vectors**.
- Margin  $\rho$**  of the separator is the width of separation between support vectors of classes.



Derivation of finding  $r$ :  
 Dotted line  $\mathbf{x}'$  is perpendicular to decision boundary so parallel to  $\mathbf{w}$ .  
 Unit vector is  $\mathbf{w}/\|\mathbf{w}\|$ , so this one is  $r\mathbf{w}/\|\mathbf{w}\|$ .  
 $\mathbf{x}' = \mathbf{x} - r\mathbf{w}/\|\mathbf{w}\|$ .  
 $\mathbf{x}'$  satisfies  $\mathbf{w}^T \mathbf{x}' + b = 0$ .  
 So  $\mathbf{w}^T (\mathbf{x} - r\mathbf{w}/\|\mathbf{w}\|) + b = 0$ .  
 Recall that  $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T \mathbf{w}}$ .  
 So, solving for  $r$  gives:  
 $r = y(\mathbf{w}^T \mathbf{x} + b)/\|\mathbf{w}\|$

7

## Linear SVM Mathematically

The linearly separable case

- Assume that all data is at least distance 1 from the hyperplane, then the following two constraints follow for a training set  $\{(\mathbf{x}_i, y_i)\}$

$$\mathbf{w}^T \mathbf{x}_i + b \geq 1 \quad \text{if } y_i = 1$$

$$\mathbf{w}^T \mathbf{x}_i + b \leq -1 \quad \text{if } y_i = -1$$

- For support vectors, the inequality becomes an equality
- Then, since each example's distance from the hyperplane is

$$r = y \frac{\mathbf{w}^T \mathbf{x} + b}{\|\mathbf{w}\|}$$

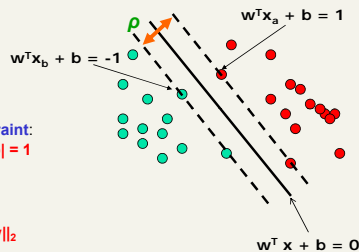
- The margin is:

$$\rho = \frac{2}{\|\mathbf{w}\|}$$

8

## Linear Support Vector Machine (SVM)

- Hyperplane**  
 $\mathbf{w}^T \mathbf{x} + b = 0$
- Extra scale constraint:**  
 $\min_{i=1, \dots, n} |\mathbf{w}^T \mathbf{x}_i + b| = 1$
- This implies:  
 $\mathbf{w}^T (\mathbf{x}_a - \mathbf{x}_b) = 2$   
 $\rho = \|\mathbf{x}_a - \mathbf{x}_b\|_2 = 2/\|\mathbf{w}\|_2$



9

## Linear SVMs Mathematically (cont.)

- Then we can formulate the *quadratic optimization problem*:

Find  $\mathbf{w}$  and  $b$  such that

$$\rho = \frac{2}{\|\mathbf{w}\|} \text{ is maximized; and for all } \{(\mathbf{x}_i, y_i)\}$$

$$\mathbf{w}^T \mathbf{x}_i + b \geq 1 \text{ if } y_i = 1; \quad \mathbf{w}^T \mathbf{x}_i + b \leq -1 \text{ if } y_i = -1$$

- A better formulation ( $\min \|\mathbf{w}\| = \max 1/\|\mathbf{w}\|$ ):

Find  $\mathbf{w}$  and  $b$  such that

$$\Phi(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} \text{ is minimized;}$$

$$\text{and for all } \{(\mathbf{x}_i, y_i)\}: \quad y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1$$

10

## Solving the Optimization Problem

Find  $\mathbf{w}$  and  $b$  such that  
 $\Phi(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w}$  is minimized;  
 and for all  $\{(\mathbf{x}_i, y_i)\}: y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1$

- This is now optimizing a *quadratic* function subject to *linear* constraints
- Quadratic optimization problems are a well-known class of mathematical programming problem, and many (intricate) algorithms exist for solving them (with many special ones built for SVMs)
- The solution involves constructing a *dual problem* where a *Lagrange multiplier*  $\alpha_i$  is associated with every constraint in the primary problem:

Find  $\alpha_1, \dots, \alpha_N$  such that  
 $Q(\boldsymbol{\alpha}) = \sum \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$  is maximized and  
 (1)  $\sum \alpha_i y_i = 0$   
 (2)  $\alpha_i \geq 0$  for all  $\alpha_i$

11

## The Optimization Problem Solution

- The solution has the form:

$$\mathbf{w} = \sum \alpha_i y_i \mathbf{x}_i \quad b = y_k - \mathbf{w}^T \mathbf{x}_k \text{ for any } \mathbf{x}_k \text{ such that } \alpha_k \neq 0$$

- Each non-zero  $\alpha_i$  indicates that corresponding  $\mathbf{x}_i$  is a support vector.
- Then the classifying function will have the form:

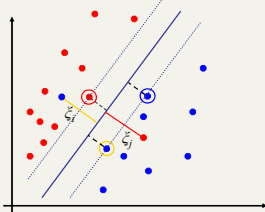
$$f(\mathbf{x}) = \sum \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b$$

- Notice that it relies on an *inner product* between the test point  $\mathbf{x}$  and the support vectors  $\mathbf{x}_i$  – we will return to this later.
- Also keep in mind that solving the optimization problem involved computing the inner products  $\mathbf{x}_i^T \mathbf{x}_j$  between all pairs of training points.

12

## Soft Margin Classification

- If the training data is not linearly separable, *slack variables*  $\xi_i$  can be added to allow misclassification of difficult or noisy examples.
- Allow some errors
  - Let some points be moved to where they belong, at a cost
- Still, try to minimize training set errors, and to place hyperplane "far" from each class (large margin)



15.2.1

## Soft Margin Classification Mathematically

- The old formulation:

Find  $w$  and  $b$  such that  $\Phi(w) = \frac{1}{2} w^T w$  is minimized and for all  $\{(x_i, y_i)\}$   $y_i (w^T x_i + b) \geq 1$

- The new formulation incorporating slack variables:

Find  $w$  and  $b$  such that  $\Phi(w) = \frac{1}{2} w^T w + C \sum \xi_i$  is minimized and for all  $\{(x_i, y_i)\}$   $y_i (w^T x_i + b) \geq 1 - \xi_i$  and  $\xi_i \geq 0$  for all  $i$

- Parameter  $C$  can be viewed as a way to control overfitting – a regularization term

14

## Soft Margin Classification – Solution

- The dual problem for soft margin classification:

Find  $a_1, \dots, a_N$  such that  $Q(\alpha) = \sum a_i - \frac{1}{2} \sum \sum a_i a_j y_i y_j x_i^T x_j$  is maximized and  
 (1)  $\sum a_i y_i = 0$   
 (2)  $0 \leq a_i \leq C$  for all  $a_i$

- Neither slack variables  $\xi_i$  nor their Lagrange multipliers appear in the dual problem!
- Again,  $x_i$  with non-zero  $a_i$  will be support vectors.
- Solution to the dual problem is:

$w = \sum a_i y_i x_i$   
 $B = y_k (1 - \xi_k) - w^T x_k$  where  $k = \operatorname{argmax}_k a_k$

$w$  is not needed explicitly for classification!

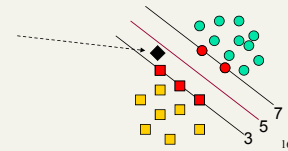
$f(x) = \sum a_i y_i x_i^T x + b$

15

## Classification with SVMs

- Given a new point  $x$ , we can score its projection onto the hyperplane normal:
  - I.e., compute score:  $w^T x + b = \sum a_i y_i x_i^T x + b$
  - Can set confidence threshold  $t$ .

Score  $> t$  yes  
 Score  $< -t$  no  
 Else: don't know



16

## Linear SVMs: Summary

- The classifier is a *separating hyperplane*.
- Most "important" training points are support vectors; they define the hyperplane.
- Quadratic optimization algorithms can identify which training points  $x_i$  are support vectors with non-zero Lagrangian multipliers  $a_i$ .
- Both in the dual formulation of the problem and in the solution training points appear only inside inner products:

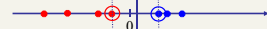
Find  $a_1, \dots, a_N$  such that  $Q(\alpha) = \sum a_i - \frac{1}{2} \sum \sum a_i a_j y_i y_j x_i^T x_j$  is maximized and  
 (1)  $\sum a_i y_i = 0$   
 (2)  $0 \leq a_i \leq C$  for all  $a_i$

$f(x) = \sum a_i y_i x_i^T x + b$

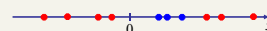
17

## Non-linear SVMs

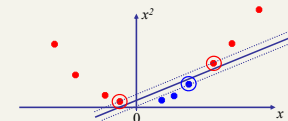
- Datasets that are linearly separable (with some noise) work out great:



- But what are we going to do if the dataset is just too hard?



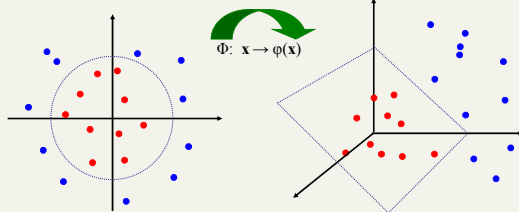
- How about ... mapping data to a higher-dimensional space:



15.2.3

## Non-linear SVMs: Feature spaces

- General idea: the original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:



19

## The "Kernel Trick"

- The linear classifier relies on an inner product between vectors  $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
- If every datapoint is mapped into high-dimensional space via some transformation  $\Phi: \mathbf{x} \rightarrow \Phi(\mathbf{x})$ , the inner product becomes:
 
$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$$
- A *kernel function* is some function that corresponds to an inner product in some expanded feature space.
- Example:  
2-dimensional vectors  $\mathbf{x} = [x_1, x_2]$ : let  $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^2$ .  
Need to show that  $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$ :
 
$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}_j) &= (1 + \mathbf{x}_i^T \mathbf{x}_j)^2 = 1 + x_{i1}^2 x_{j1}^2 + 2 x_{i1} x_{j1} x_{i2} x_{j2} + x_{i2}^2 x_{j2}^2 + 2 x_{i1} x_{j1} + 2 x_{i2} x_{j2} \\ &= [1 \ x_{i1}^2 \ \sqrt{2} \ x_{i1} x_{i2} \ x_{i2}^2 \ \sqrt{2} x_{i1} \ \sqrt{2} x_{i2}]^T [1 \ x_{j1}^2 \ \sqrt{2} \ x_{j1} x_{j2} \ x_{j2}^2 \ \sqrt{2} x_{j1} \ \sqrt{2} x_{j2}] \\ &= \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) \quad \text{where } \Phi(\mathbf{x}) = [1 \ x_1^2 \ \sqrt{2} \ x_1 x_2 \ x_2^2 \ \sqrt{2} x_1 \ \sqrt{2} x_2] \end{aligned}$$

## Kernels

- Why use kernels?
  - Make non-separable problem separable.
  - Map data into better representational space
- Common kernels
  - Linear
  - Polynomial  $K(\mathbf{x}, \mathbf{z}) = (1 + \mathbf{x}^T \mathbf{z})^d$ 
    - Gives feature conjunctions
  - Radial basis function (infinite dimensional space)
 
$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2}$$
- Haven't been very useful in text classification

21

## Evaluation: Classic Reuters Data Set

- Most (over)used data set
- 21578 documents
- 9603 training, 3299 test articles (ModApte split)
- 118 categories
  - An article can be in more than one category
  - Learn 118 binary category distinctions
- Average document: about 90 types, 200 tokens
- Average number of classes assigned
  - 1.24 for docs with at least one category
- Only about 10 out of 118 categories are large

Common categories	• Earn (2877, 1087)	• Trade (369, 119)
(#train, #test)	• Acquisitions (1650, 179)	• Interest (347, 131)
	• Money-fx (538, 179)	• Ship (197, 89)
	• Grain (433, 149)	• Wheat (212, 71)
	• Crude (389, 189)	• Corn (182, 56)

22

## Reuters Text Categorization data set (Reuters-21578) document

```
<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET"
OLDID="12981" NEWID="798">
<DATE> 2-MAR-1987 16:51:43.42</DATE>
<TOPICS><D>livestock</D><D>hog</D></TOPICS>
<TITLE>AMERICAN PORK CONGRESS KICKS OFF TOMORROW</TITLE>
<DATELINE> CHICAGO, March 2 - </DATELINE><BODY>The American Pork Congress
kicks off tomorrow, March 3, in Indianapolis with 160 of the nations pork producers from 44
member states determining industry positions on a number of issues, according to the National Pork
Producers Council, NPPC.
Delegates to the three day Congress will be considering 26 resolutions concerning various issues,
including the future direction of farm policy and the tax law as it applies to the agriculture sector.
The delegates will also debate whether to endorse concepts of a national PRV (pseudorabies virus)
control and eradication program, the NPPC said.
A large trade show, in conjunction with the congress, will feature the latest in technology in all
areas of the industry, the NPPC added. Reuters
&#3;</BODY></TEXT></REUTERS>
```

23

## Per class evaluation measures

- Recall: Fraction of docs in class  $i$  classified correctly:
 
$$\frac{c_{ii}}{\sum_j c_{ij}}$$
- Precision: Fraction of docs assigned class  $i$  that are actually about class  $i$ :
 
$$\frac{c_{ii}}{\sum_j c_{ji}}$$
- "Correct rate": (1 - error rate) Fraction of docs classified correctly:
 
$$\frac{\sum_i c_{ii}}{\sum_j \sum_i c_{ij}}$$

24

## 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.

25

## Micro- vs. Macro-Averaging: Example

Class 1                      Class 2                      Micro.Av. Table

	Truth: yes	Truth: no		Truth: yes	Truth: no		Truth: yes	Truth: no
Classifier: yes	10	10	Classifier: yes	90	10	Classifier: yes	100	20
Classifier: no	10	970	Classifier: no	10	890	Classifier: no	20	1860

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

26

(a)

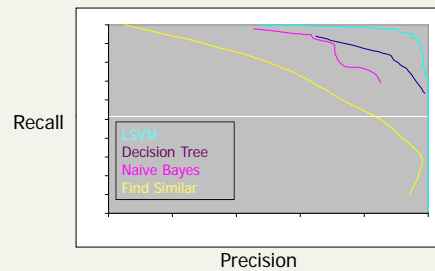
	NB	Rocchio	kNN	SVM
micro-avg-L (90 classes)	80	85	86	89
macro-avg (90 classes)	47	59	60	60

(b)

	NB	Rocchio	kNN	trees	SVM
earn	96	93	97	98	98
acq	88	65	92	90	94
money-fx	57	47	78	66	75
grain	79	68	82	85	95
crude	80	70	86	85	89
trade	64	65	77	73	76
interest	65	63	74	67	78
ship	85	49	79	74	86
wheat	70	69	77	93	92
corn	65	48	78	92	90
micro-avg (top 10)	82	65	82	88	92
micro-avg-D (118 classes)	75	62	n/a	n/a	87

Evaluation measure:  $F_1$

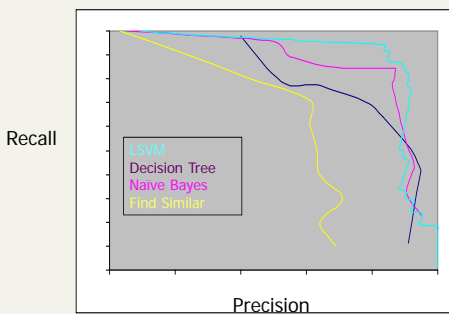
## Reuters ROC - Category Grain



**Recall:** % labeled in category among those stories that are really in category  
**Precision:** % really in category among those stories labeled in category

28

## ROC for Category - Ship



29

## Yang&Liu: SVM vs. Other Methods

Table 1: Performance summary of classifiers

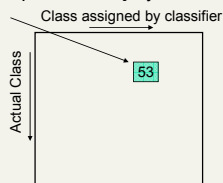
method	miR	miP	miF1	maF1	error
SVM	.8120	.9137	.8599	.5251	.00365
KNN	.8339	.8807	.8567	.5242	.00385
LSF	.8507	.8489	.8498	.5008	.00414
NNet	.7842	.8785	.8287	.3765	.00447
NB	.7688	.8245	.7956	.3886	.00544

miR = micro-avg recall; miP = micro-avg prec.;  
miF1 = micro-avg F1; maF1 = macro-avg F1.

30

## Good practice department: Confusion matrix

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



- In a perfect classification, only the diagonal has non-zero entries

31

## The Real World

P. Jackson and I. Moulinier: *Natural Language Processing for Online Applications*

- "There is no question concerning the commercial value of being able to classify documents automatically by content. There are myriad potential applications of such a capability for corporate Intranets, government departments, and Internet publishers"
- "Understanding the data is one of the keys to successful categorization, yet this is an area in which most categorization tool vendors are extremely weak. Many of the 'one size fits all' tools on the market have not been tested on a wide range of content types."

32

## The Real World

- Gee, I'm building a text classifier for real, now!
- What should I do?
- How much training data do you have?
  - None
  - Very little
  - Quite a lot
  - A huge amount and its growing

33

## Manually written rules

- No training data, adequate editorial staff?
- Never forget the hand-written rules solution!
  - If (wheat or grain) and not (whole or bread) then
    - Categorize as grain
- In practice, rules get a lot bigger than this
  - Can also be phrased using tf or tf.idf weights
- With careful crafting (human tuning on development data) performance is high:
  - Construe: 94% recall, 84% precision over 675 categories (Hayes and Weinstein 1990)
- Amount of work required is huge
  - Estimate 2 days per class ... plus maintenance

34

## Very little data?

- If you're just doing supervised classification, you should stick to something high bias
  - There are theoretical results that Naïve Bayes should do well in such circumstances (Ng and Jordan 2002 NIPS)
- The interesting theoretical answer is to explore semi-supervised training methods:
  - Bootstrapping, EM over unlabeled documents, ...
- The practical answer is to get more labeled data as soon as you can
  - How can you insert yourself into a process where humans will be willing to label data for you??

35

## A reasonable amount of data?

- Perfect!
- We can use all our clever classifiers
- Roll out the SVM!
- But if you are using an SVM/NB etc., you should probably be prepared with the "hybrid" solution where there is a Boolean overlay
  - Or else to use user-interpretable Boolean-like models like decision trees
  - Users like to hack, and management likes to be able to implement quick fixes immediately

36

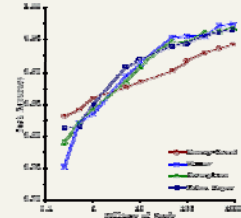
## A huge amount of data?

- This is great in theory for doing accurate classification...
- But it could easily mean that expensive methods like SVMs (train time) or kNN (test time) are quite impractical
- Naive Bayes can come back into its own again!
  - Or other advanced methods with linear training/test complexity like regularized logistic regression (though much more expensive to train)

37

## A huge amount of data?

- With enough data the choice of classifier may not matter much, and the best choice may be unclear
  - Data: Brill and Banko on context-sensitive spelling correction
- But the fact that you have to keep doubling your data to improve performance is a little unpleasant



38

## How many categories?

- A few (well separated ones?)
  - Easy!
- A zillion closely related ones?
  - Think: Yahoo! Directory, Library of Congress classification, legal applications
  - Quickly gets difficult!
    - Classifier combination is always a useful technique
      - Voting, bagging, or boosting multiple classifiers
    - Much literature on hierarchical classification
      - Mileage fairly unclear
    - May need a hybrid automatic/manual solution

39

## How can one tweak performance?

- Aim to exploit any domain-specific useful features that give special meanings or that zone the data
  - E.g., an author byline or mail headers
- Aim to collapse things that would be treated as different but shouldn't be.
  - E.g., part numbers, chemical formulas

40

## Does putting in "hacks" help?

- You bet!
- You can get a lot of value by differentially weighting contributions from different document zones:
  - Upweighting title words helps (Cohen & Singer 1996)
    - Doubling the weighting on the title words is a good rule of thumb
  - Upweighting the first sentence of each paragraph helps (Murata, 1999)
  - Upweighting sentences that contain title words helps (Ko *et al.*, 2002)

41

## Two techniques for zones

1. Have a completely separate set of features/parameters for different zones like the title
  2. Use the same features (pooling/tying their parameters) across zones, but upweight the contribution of different zones
- Commonly the second method is more successful: it costs you nothing in terms of sparsifying the data, but can give a very useful performance boost
    - Which is best is a contingent fact about the data

42

## Text Summarization techniques in text classification

- Text Summarization: Process of extracting key pieces from text, normally by features on sentences reflecting position and content
- Much of this work can be used to suggest weightings for terms in text categorization
  - See: Kolcz, Prabhakar, and Kalita, CIKM 2001: Summarization as feature selection for text categorization
  - Categorizing purely with title,
  - Categorizing with first paragraph only
  - Categorizing with paragraph with most keywords
  - Categorizing with first and last paragraphs, etc.

43

## Does stemming/lowercasing/... help?

- As always it's hard to tell, and empirical evaluation is normally the gold standard
- But note that the role of tools like stemming is rather different for TextCat vs. IR:
  - For IR, you often want to collapse forms of the verb *oxygenate* and *oxygenation*, since all of those documents will be relevant to a query for *oxygenation*
  - For TextCat, with sufficient training data, stemming *does no good*. It only helps in compensating for data sparseness (which can be severe in TextCat applications). *Overly aggressive stemming can easily degrade performance.*

44

## Measuring Classification Figures of Merit

- Not just accuracy; in the real world, there are economic measures:
  - Your choices are:
    - Do no classification
      - That has a cost (hard to compute)
    - Do it all manually
      - Has an easy to compute cost if doing it like that now
    - Do it all with an automatic classifier
      - Mistakes have a cost
    - Do it with a combination of automatic classification and manual review of uncertain/difficult/"new" cases
  - Commonly the last method is most cost efficient and is adopted

45

## A common problem: 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.
  - Can favor simpler models like Naïve Bayes
- **Feature selection: can be bad in protecting against concept drift**

46

## Summary

- Support vector machines (SVM)
  - Choose hyperplane based on support vectors
    - Support vector = "critical" point close to decision boundary
  - (Degree-1) SVMs are linear classifiers.
  - Kernels: powerful and elegant way to define similarity metric
  - Perhaps best performing text classifier
    - But there are other methods that perform about as well as SVM, such as regularized logistic regression (Zhang & Oles 2001)
  - Partly popular due to availability of SVMlight
    - SVMlight is accurate and fast – and free (for research)
    - Now lots of software: libsvm, TinySVM, ....
  - Comparative evaluation of methods
  - Real world: exploit domain specific structure!

47

## Resources

- A Tutorial on Support Vector Machines for Pattern Recognition (1998) Christopher J. C. Burges
- S. T. Dumais, Using SVMs for text categorization, IEEE Intelligent Systems, 13(4), Jul/Aug 1998
- S. T. Dumais, J. Platt, D. Heckerman and M. Sahami. 1998. Inductive learning algorithms and representations for text categorization. *CIKM '98*, pp. 148-155.
- A re-examination of text categorization methods (1999) Yiming Yang, Xin Liu 22nd Annual International SIGIR
- Tong Zhang, Frank J. Oles: Text Categorization Based on Regularized Linear Classification Methods. *Information Retrieval* 4(1): 5-31 (2001)
- Trevor Hastie, Robert Tibshirani and Jerome Friedman, "Elements of Statistical Learning: Data Mining, Inference and Prediction" Springer-Verlag, New York.
- 'Classic' Reuters data set: <http://www.daviddlewis.com/resources/testcollections/reuters21578/>
- T. Joachims, *Learning to Classify Text using Support Vector Machines*. Kluwer, 2002.
- Fan Li, Yiming Yang: A Loss Function Analysis for Classification Methods in Text Categorization. *ICML 2003*: 472-479.

48