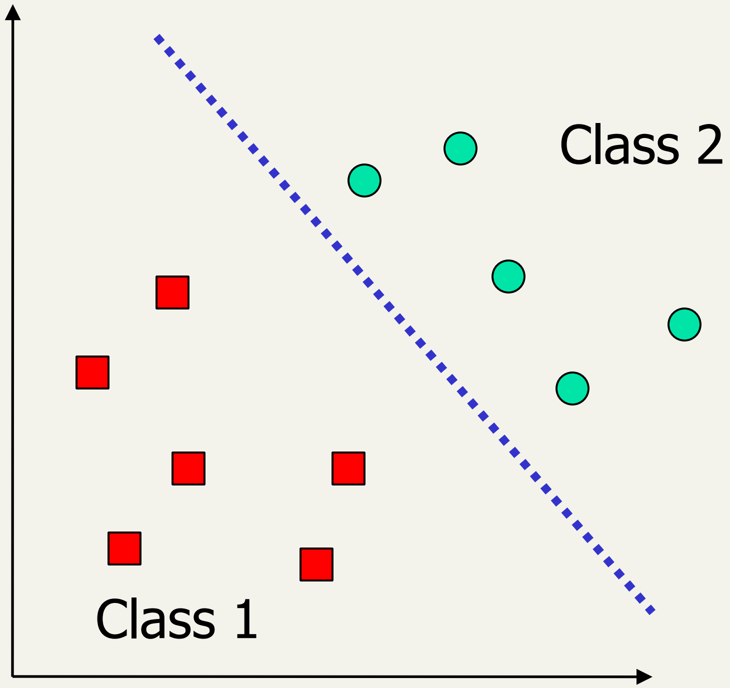
Support Vector Machines

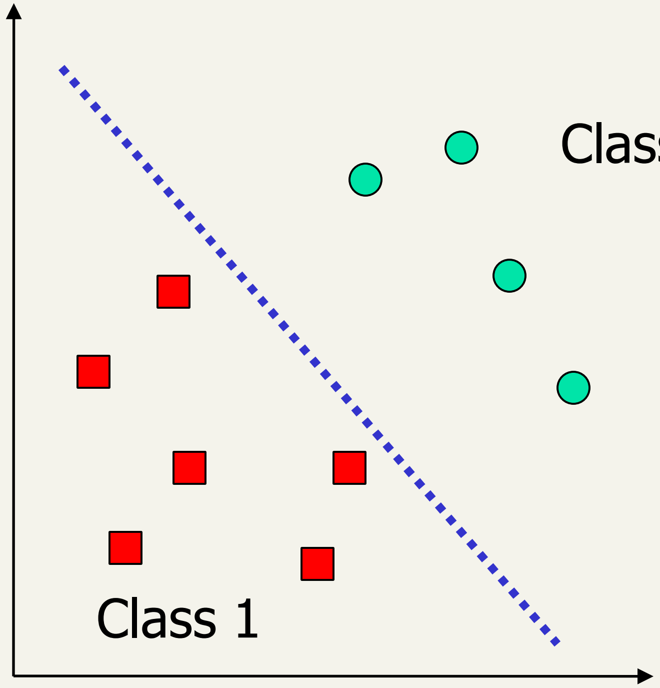
290N, 2015

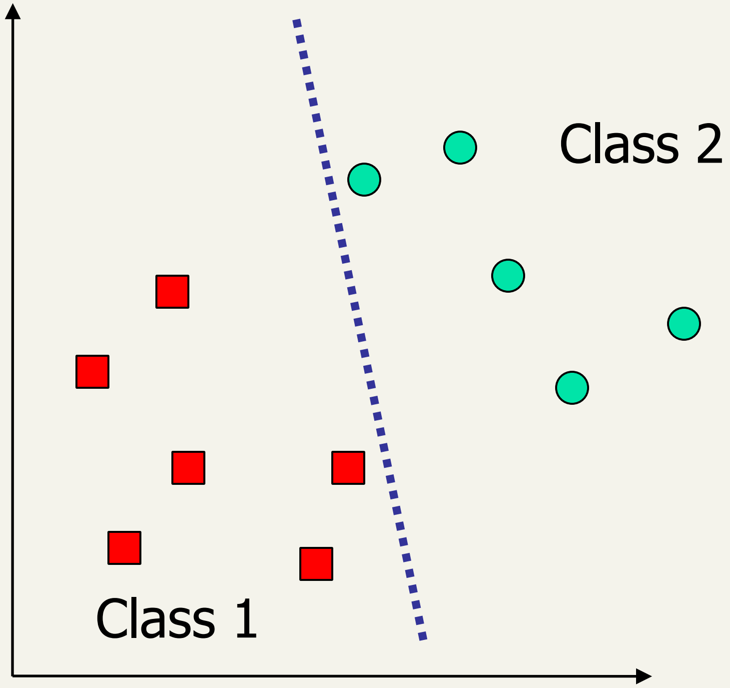
Two Class Problem: Linear Separable Case with a Hyperplane



- Many decision boundaries can separate these two classes using a hyperplane.
- Which one should we choose?

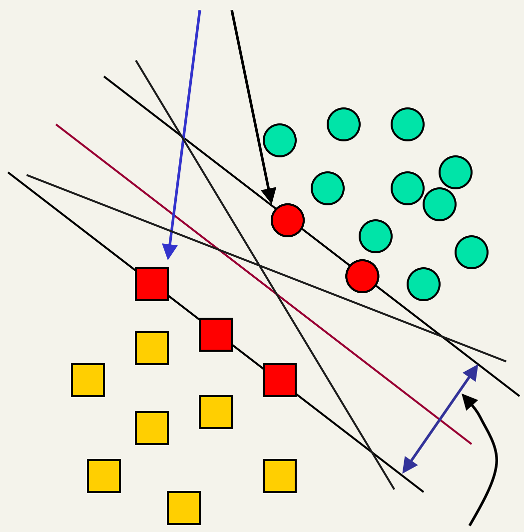
Example of Bad Decision Boundaries





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*.
- *Quadratic programming* problem

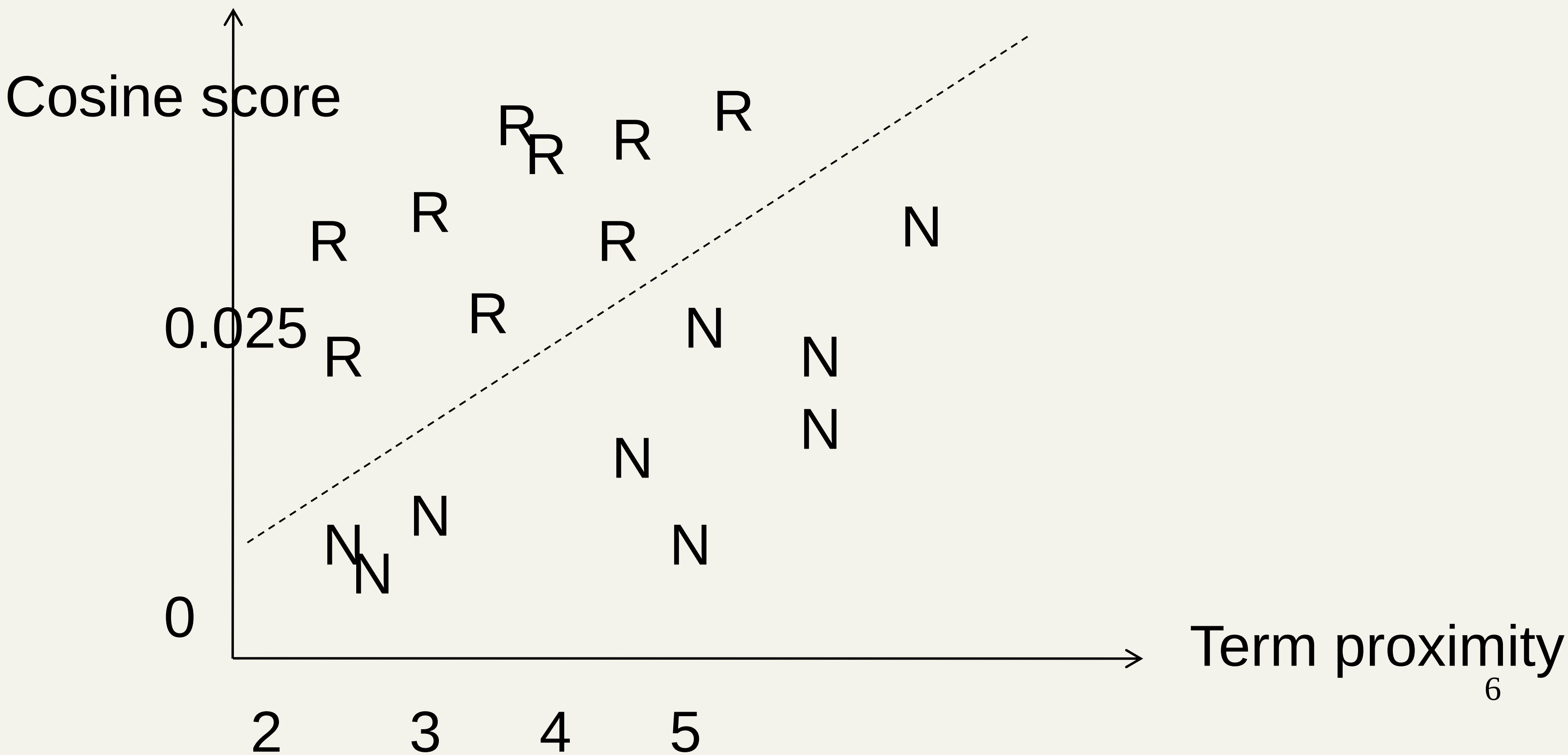


Training examples for document ranking

Two ranking signals are used (Cosine text similarity score, proximity of term appearance window)					
Example	DocID	Query	Cosine score	ω	Judgment
Φ_1	37	linux operating system	0.032	3	*relevant*
Φ_2	37	penguin logo	0.02	4	*nonrelevant*
Φ_3	238	operating system	0.043	2	*relevant*
Φ_4	238	runtime environment	0.004	2	*nonrelevant*
Φ_5	1741	kernel layer	0.022	3	*relevant*
Φ_6	2094	device driver	0.03	2	*relevant*
Φ_7	3191	device driver	0.027	5	*nonrelevant*
...	...	...	...		

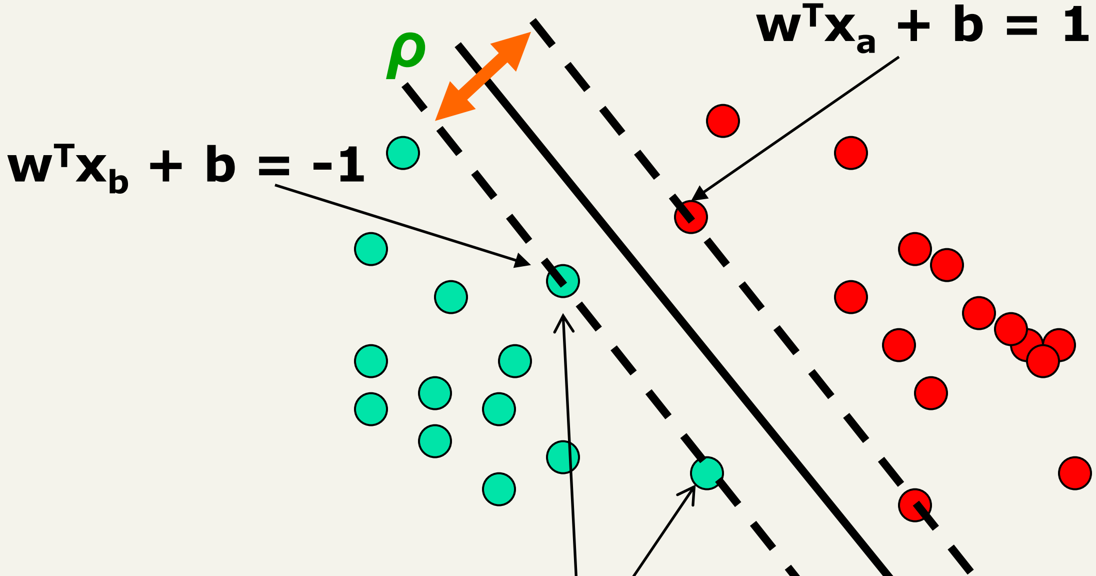
Proposed scoring function for ranking

$$Score(d,q) = Score(\alpha,\omega) = a\alpha + b\omega + c,$$



Formalization

- w: weight coefficients
- x_i: data point i
- y_i: class result of data point i (+1 or -1)
- Classifier is: $f(x_i) = \text{sign}(w^T x_i + b)$



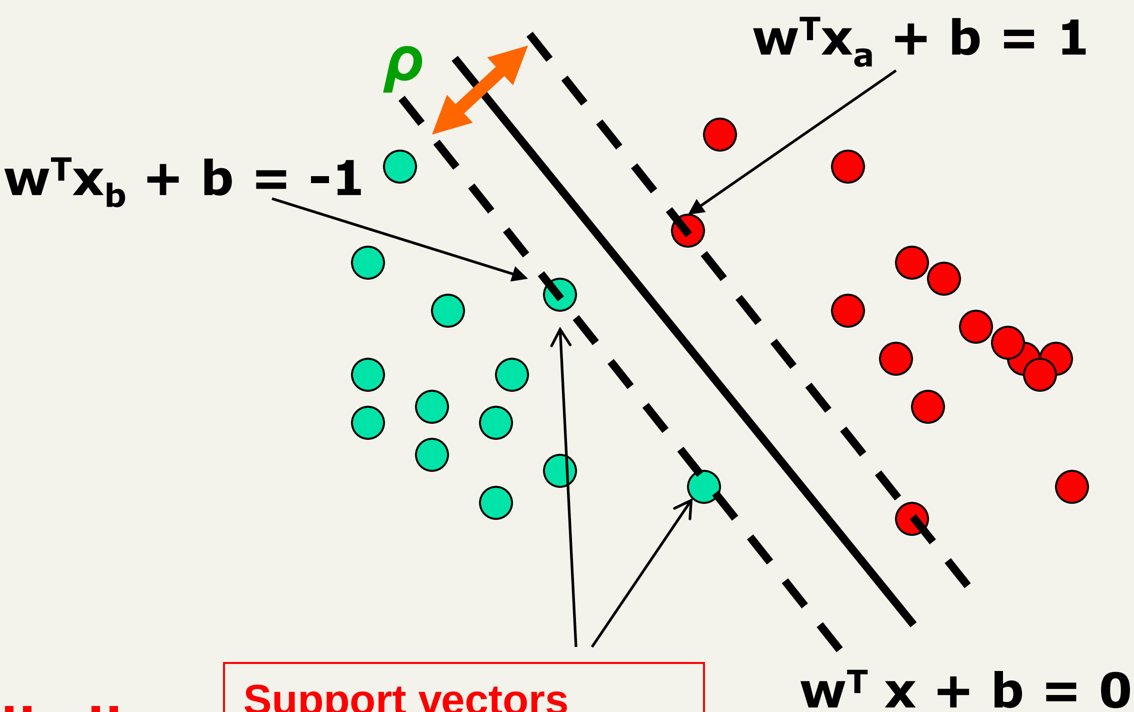
Linear Support Vector Machine (SVM)

- **Hyperplane**

$w^T x + b = 0$

$w^T x + b = 1$

$w^T x + b = -1$



- $\rho = ||x_a - x_b||_2 = \mathbf{2/||w||_2}$

Support vectors
datapoints that the margin
pushes up against

Linear SVM Mathematically

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

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

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

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

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

- The margin of dataset is:

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

The Optimization Problem

- Let $\{x_1, \ldots, x_n\}$ be our data set and let $y_i \in \{1,-1\}$ be the class label of x_i
- The decision boundary should classify all points correctly $\Rightarrow$
- A constrained optimization problem

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2$$

- $||\mathbf{w}||^2 = \mathbf{w}^T\mathbf{w}$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad \forall i$$

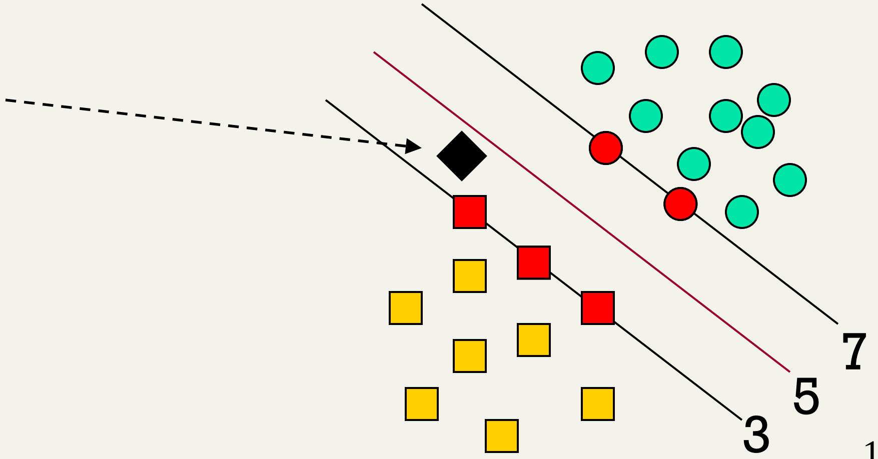
Classification with SVMs

- Given a new point (x_1,x_2), we can score its projection onto the hyperplane normal:
  - In 2 dims: score = $w_1x_1+w_2x_2+b$.
    - I.e., compute score: $wx + b = \Sigma\alpha_i y_i \mathbf{x_i}^T\mathbf{x} + b$
  - Set confidence threshold t.

Score > t: yes

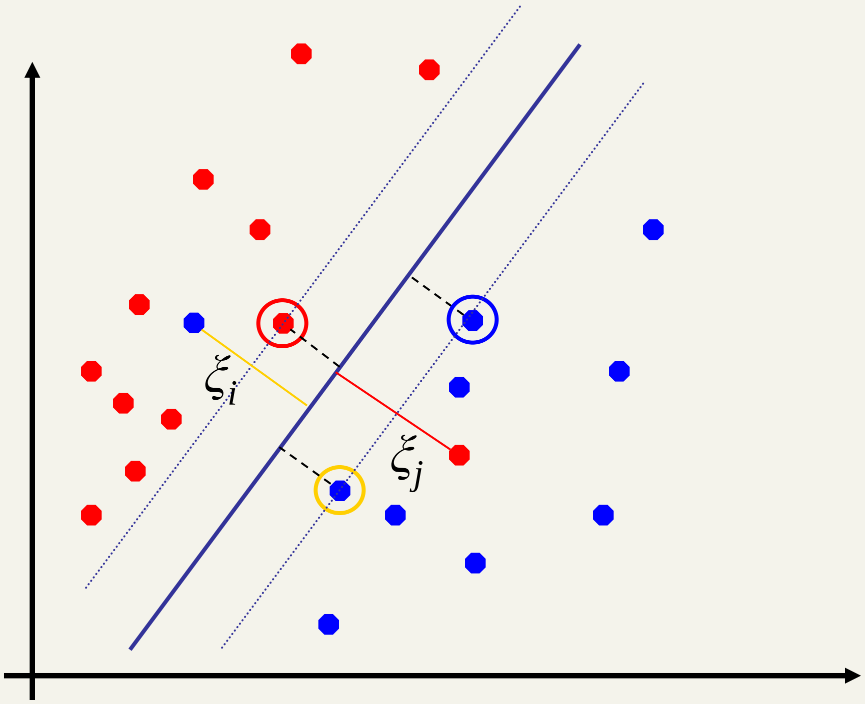
Score < -t: no

Else: don't know



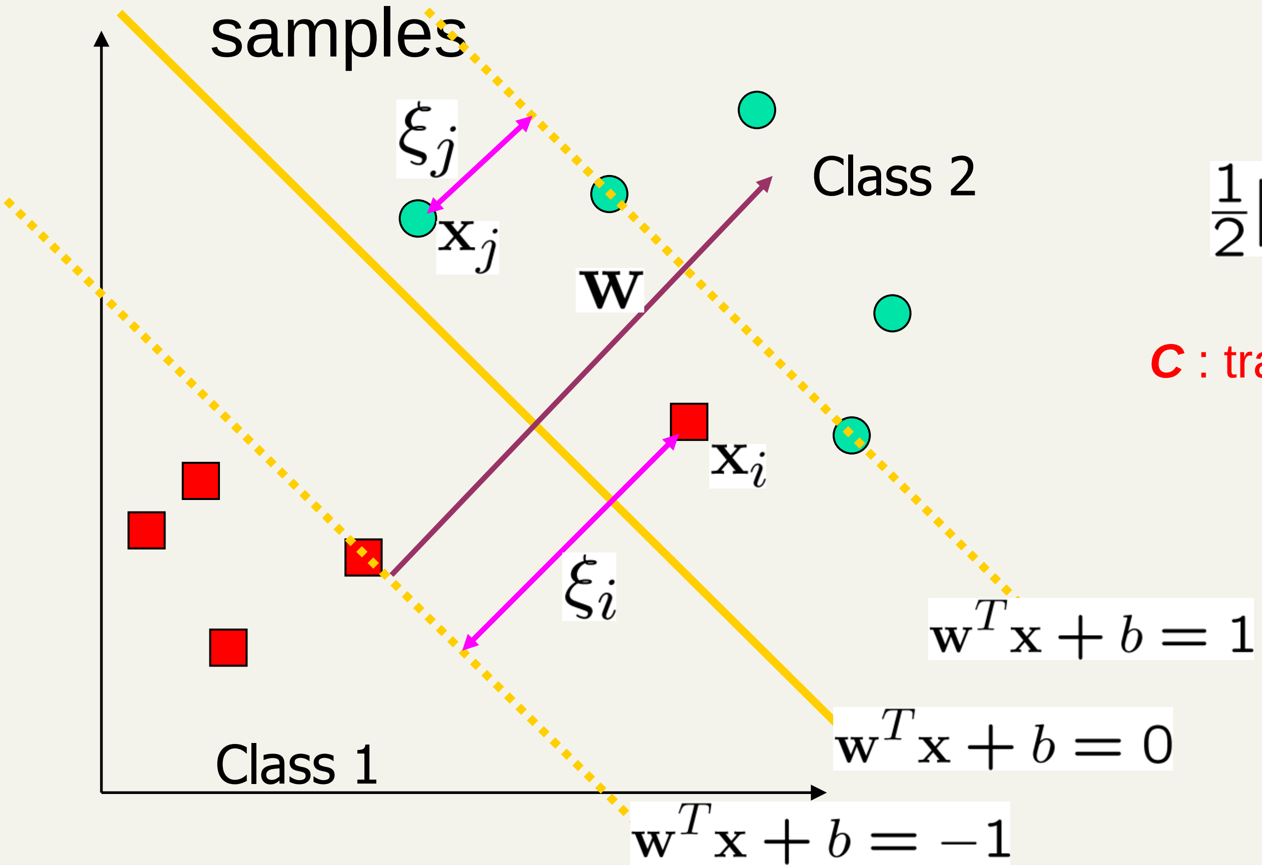
Soft Margin Classification

- If the training set is not linearly separable, *slack variables* ξ_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)



Soft margin

- We allow "error" ξ_i in classification; it is based on the output of the discriminant function $\mathbf{w}^T\mathbf{x}+b$
- ξ_i approximates the number of misclassified samples



New objective function:

$$\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$$

C : tradeoff parameter between error and margin; chosen by the user; large C means a higher penalty to errors

Soft Margin Classification Mathematically

- The old formulation:

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

$\mathbf{\Phi}(\mathbf{w}) = ½\, \mathbf{w}^T\mathbf{w}$ is minimized and for all $\{(\mathbf{x_i}, y_i)\}$

$y_i\, (\mathbf{w}^\mathbf{T}\mathbf{x_i} + \mathrm{b}) \geq 1$

- The new formulation incorporating slack variables:

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

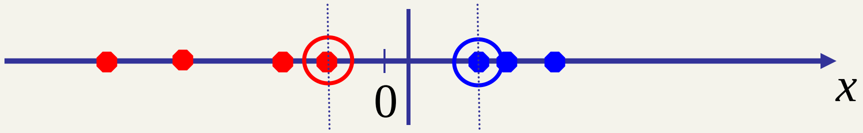
$\mathbf{\Phi}(\mathbf{w}) = ½\, \mathbf{w}^T\mathbf{w} + C\Sigma\xi_i$ is minimized and for all $\{(\mathbf{x_i}, y_i)\}$

$y_i\, (\mathbf{w}^\mathbf{T}\mathbf{x_i} + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all i

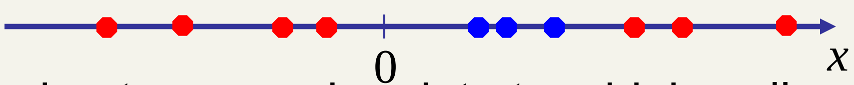
- Parameter *C* can be viewed as a way to control overfitting – a regularization term

Non-linear SVMs

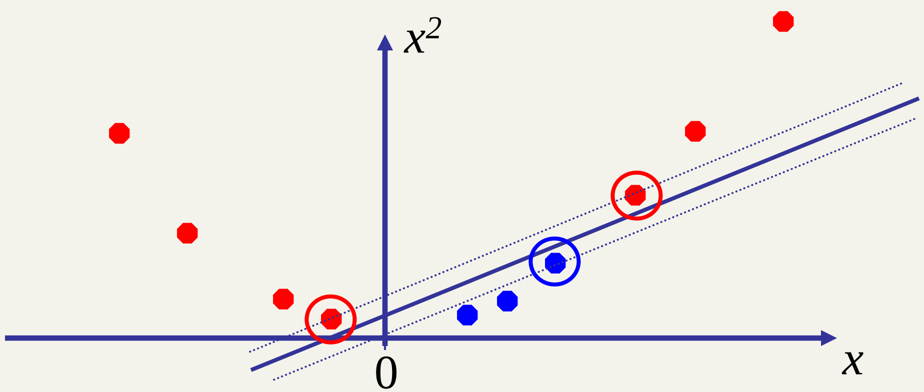
- Datasets that are linearly separable (with some noise) work out great:



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

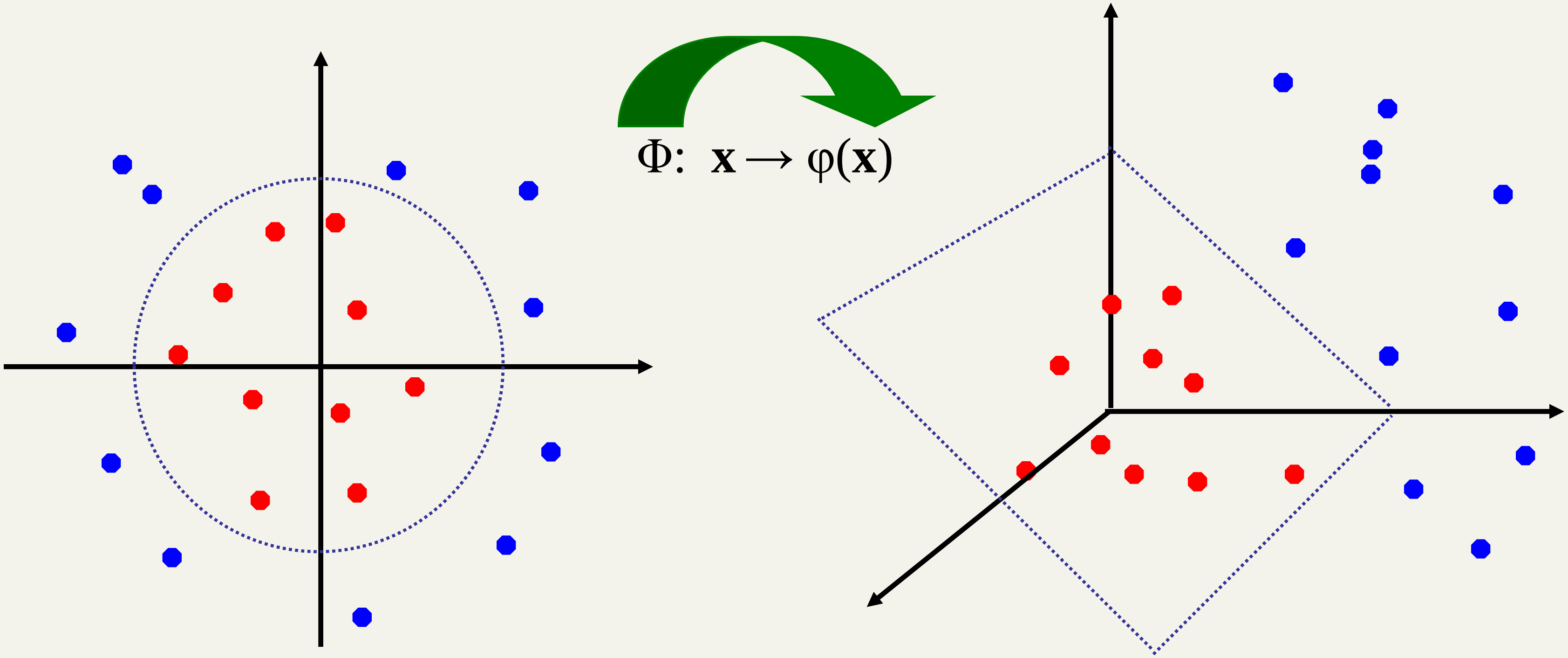


- How about … mapping data to a higher-dimensional space:



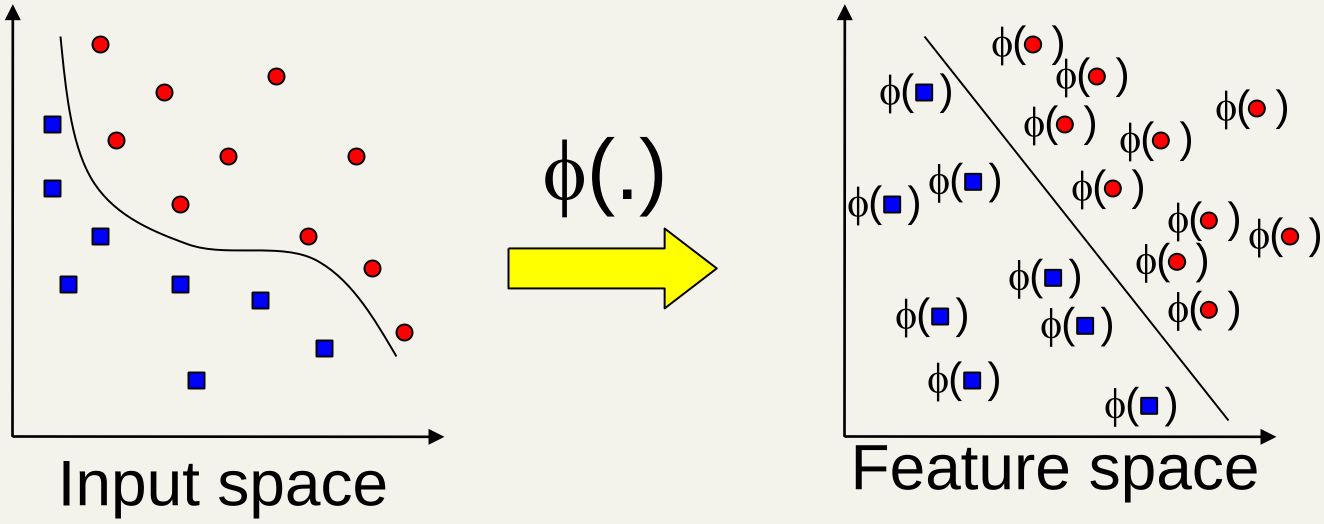
Non-linear SVMs: Feature spaces

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



Transformation to Feature Space

- "Kernel tricks"
  - Make non-separable problem separable.
  - Map data into better representational space



Modification Due to Kernel Function

- Change all inner products to kernel functions
- For training,

Original

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

With kernel function

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

Example Transformation

- Consider the following transformation

$$\phi(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

$$\phi(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}) = (1, \sqrt{2}y_1, \sqrt{2}y_2, y_1^2, y_2^2, \sqrt{2}y_1y_2)$$

$$\begin{aligned} \langle \phi(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}), \phi(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}) \rangle &= (1 + x_1y_1 + x_2y_2)^2 \\ &= K(\mathbf{x}, \mathbf{y}) \end{aligned}$$

- Define the kernel function *K* (**x**,**y**) as

$$K(\mathbf{x}, \mathbf{y}) = (1 + x_1y_1 + x_2y_2)^2$$

- The inner product $\phi(.)\phi(.)$ can be computed by *K* without going through the map $\phi(.)$ explicitly!!!

Choosing a Kernel Function

- Active research on kernel function choices for different applications
- Examples:
  - Polynomial kernel with degree *d*

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + 1)^d$$

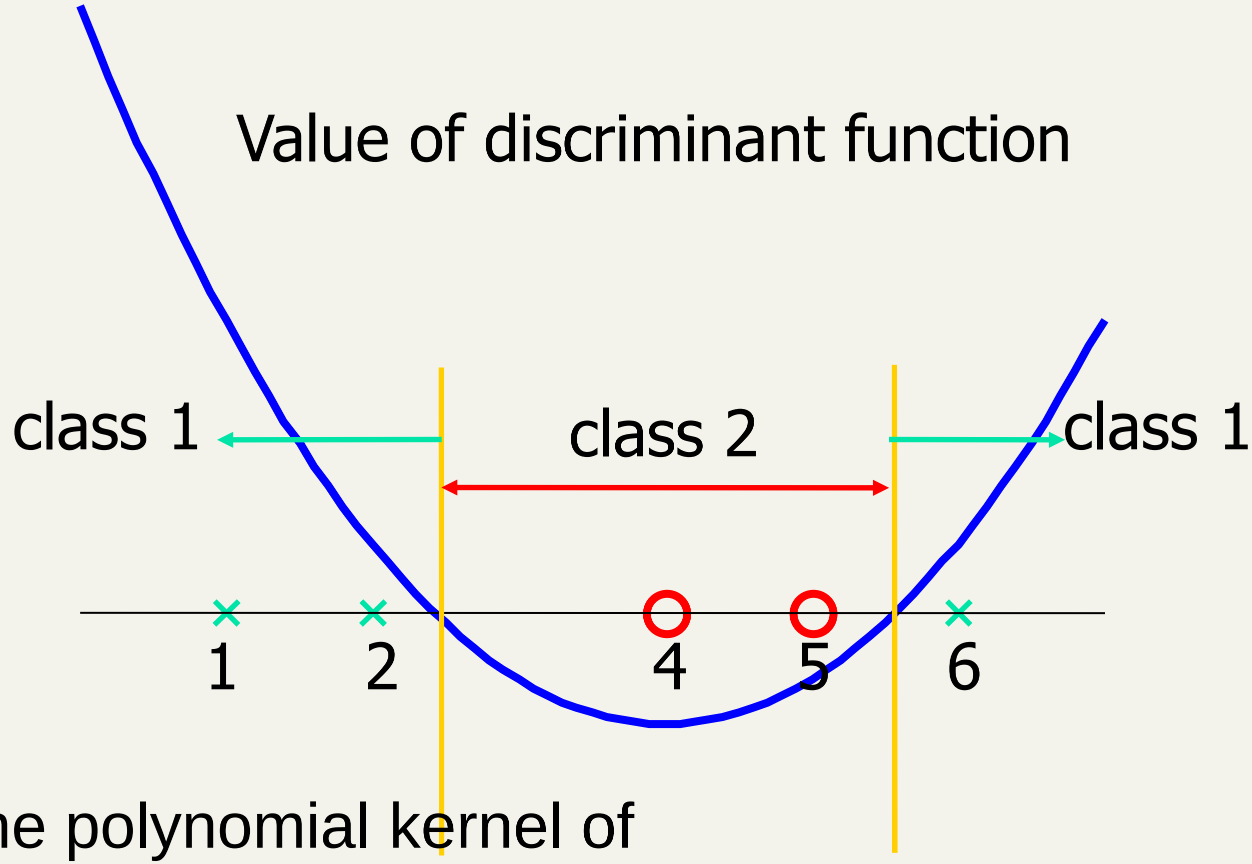
  - Radial basis function (RBF) kernel

$$k(\mathbf{x_i}, \mathbf{x_j}) = \exp(-\gamma \|\mathbf{x_i} - \mathbf{x_j}\|^2)$$

  or sometime $K(\mathbf{x}, \mathbf{y}) = \exp(-||\mathbf{x} - \mathbf{y}||^2/(2\sigma^2))$

    - Closely related to radial basis function neural networks
- In practice, a low degree polynomial kernel or RBF kernel is a good initial try

Example: 5 1D data points



We use the polynomial kernel of degree 2

$K(x,y) = (xy+1)^2$

Software

- A list of SVM implementation can be found at http://www.kernel-machines.org/software.html
- Some implementation (such as LIBSVM) can handle multi-class classification
- SVMLight is among one of the earliest implementation of SVM
- Several Matlab toolboxes for SVM are also available

Evaluation: Reuters News Data Set

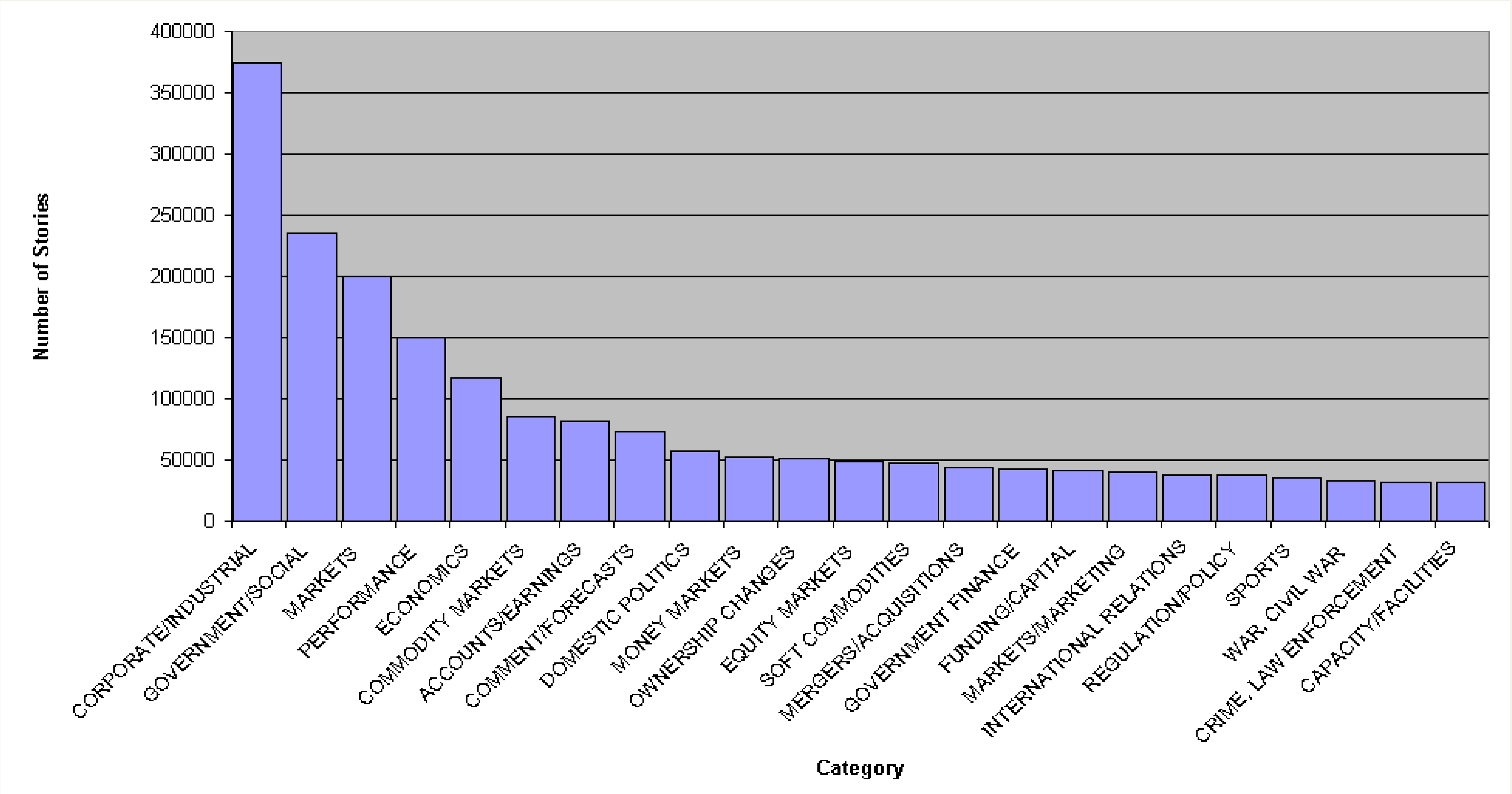
- Most (over)used data set
- 21578 documents
- 9603 training, 3299 test articles (ModApte split)
- 118 categories
  - An article can be in more than one category
  - Learn 118 binary category distinctions
- Average document: about 90 types, 200 tokens
- Average number of classes assigned
  - 1.24 for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

Common categories (#train, #test)

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

New Reuters: RCV1: 810,000 docs

- Top topics in Reuters RCV1



Dumais et al. 1998: Reuters - Accuracy

	Rocchio	NBayes	Trees	LinearSVM	
earn	92.9%	95.9%	97.8%	98.2%	
acq	64.7%	87.8%	89.7%	92.8%	
money-fx	46.7%	56.6%	66.2%	74.0%	
grain	67.5%	78.8%	85.0%	92.4%	
crude	70.1%	79.5%	85.0%	88.3%	
trade	65.1%	63.9%	72.5%	73.5%	
interest	63.4%	64.9%	67.1%	76.3%	
ship	49.2%	85.4%	74.2%	78.0%	
wheat	68.9%	69.7%	92.5%	89.7%	
corn	48.2%	65.3%	91.8%	91.1%	
Avg Top 10	64.6%	81.5%	88.4%	91.4%	
Avg All Cat	61.7%	75.2%	na	86.4%	

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

Precision: % really in category among those stories labeled in category

Break Even: (Recall + Precision) / 2

Results for Kernels (Joachims 1998)

	Bayes	Rocchio	C4.5	k-NN	SVM (poly) degree $d =$ 1	2	3	4	5	SVM (rbf) width $\gamma =$ 0.6	0.8	1.0	1.2
earn	95.9	96.1	96.1	97.3	98.2	98.4	**98.5**	98.4	98.3	**98.5**	98.5	98.4	98.3
acq	91.5	92.1	85.3	92.0	92.6	94.6	**95.2**	95.2	95.3	95.0	95.3	95.3	**95.4**
money-fx	62.9	67.6	69.4	78.2	66.9	72.5	75.4	74.9	**76.2**	74.0	75.4	**76.3**	75.9
grain	72.5	79.5	89.1	82.2	91.3	93.1	**92.4**	91.3	89.9	**93.1**	91.9	91.9	90.6
crude	81.0	81.5	75.5	85.7	86.0	87.3	88.6	**88.9**	87.8	**88.9**	89.0	88.9	88.2
trade	50.0	77.4	59.2	77.4	69.2	75.5	76.6	77.3	**77.1**	76.9	78.0	**77.8**	76.8
interest	58.0	72.5	49.1	74.0	69.8	63.3	67.9	73.1	**76.2**	74.4	75.0	**76.2**	76.1
ship	78.7	83.1	80.9	79.2	82.0	85.4	86.0	**86.5**	86.0	**85.4**	86.5	87.6	87.1
wheat	60.6	79.4	85.5	76.6	83.1	84.5	85.2	**85.9**	83.8	**85.2**	85.9	85.9	85.9
corn	47.3	62.2	87.7	77.9	86.0	86.5	85.3	**85.7**	83.9	**85.1**	85.7	85.7	84.5
microavg.	**72.0**	**79.9**	**79.4**	**82.3**	84.2	85.1	85.9	86.2	85.9	86.4	86.5	86.3	86.2
					combined: **86.0**					combined: **86.4**			

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?

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

Manually written rules

- No training data, adequate editorial staff?
- Never forget the hand-written rules solution!
  - If (wheat or grain) 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:
  - 94% recall, 84% precision over 675 categories (Hayes and Weinstein 1990)
- Amount of work required is huge
  - Estimate 2 days per class … plus maintenance

A reasonable amount of data?

- Good with 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

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
- Naïve 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)

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

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, Prabakarmurthi, and Kolita, 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.

Does stemming/lowercasing/… help?

- As always it's hard to tell
- The role of tools like stemming is slightly different for TextCat vs. IR:
  - For IR, you may want to collapse forms of the credit card/credit cards, since all of those documents will be relevant to a query for *credit card*
    - *Error happens when doing aggressively.*
    - *Avoid when there is enough data.*
  - 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.*

Measuring Classification Figures of Merit

- Not just accuracy; in the real world, there are economic measures:
  - Your choices are:
    - Do no classification
    - Do it manually
    - 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

Summary

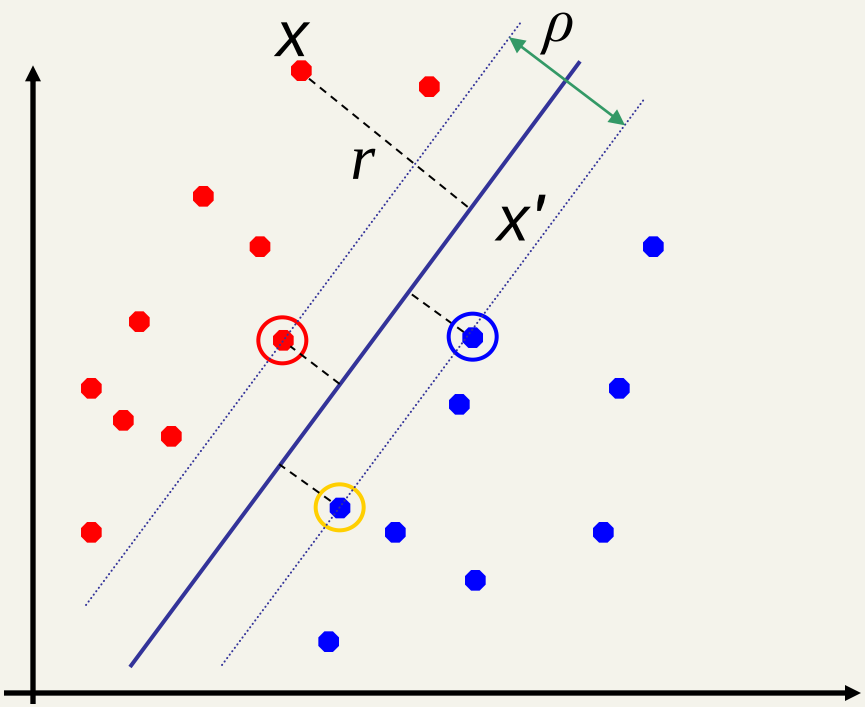
- Support vector machines (SVM)
  - Choose hyperplane based on support vectors
    - Support vector = “critical” point close to decision boundary
  - (Degree-1) SVMs are linear classifiers.
  - Kernels: powerful and elegant way to define similarity metric
  - Perhaps best performing text classifier
    - 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!

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.

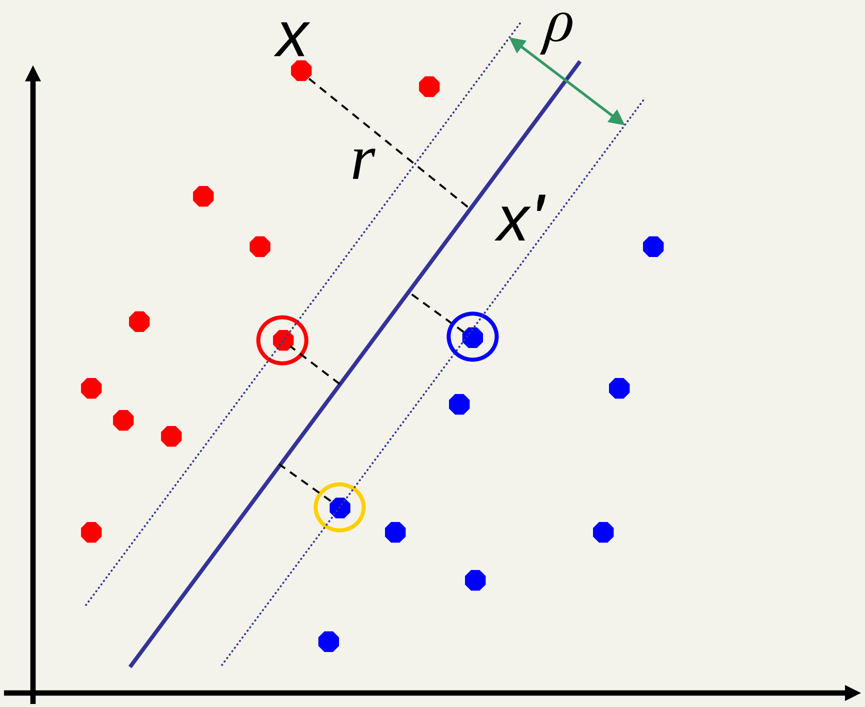
Geometric View: Margin of a point

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



Geometric View of Margin

- Distance to the separator is $r = y\dfrac{\mathbf{w}^T\mathbf{x}+b}{\|\mathbf{w}\|}$
- Let X be in line wTx+b=z. Thus (wTx+b) –(wTx'+b)=z-0 then |w| |x-x'|= |z| = y(wTx+b) thus |w| r = y(wTx+b).



Linear Support Vector Machine (SVM)

- **Hyperplane**

  $w^T x + b = 0$

- This implies:

  $w^T(x_a - x_b) = 2$

  $\rho = ||x_a - x_b||_2 = \mathbf{2/||w||_2}$