CS276A
Text Retrieval and Mining

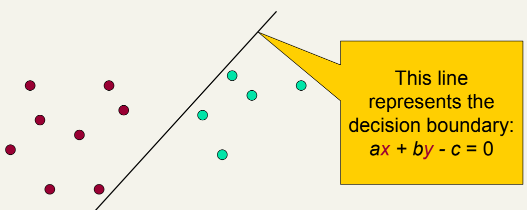
Lecture 17

[Borrows some slides from Ray Mooney]

Recap of the last lecture

- (Re)Introduction to Text Classification
- 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; too simple
  - Decision Trees
    - Pick out hyperboxes; nonlinear; use just a few features

Recall: Linear classifiers

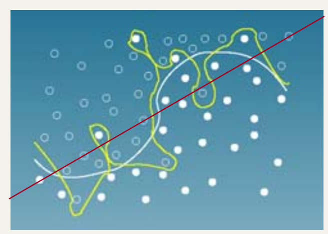


Want to find a,b,c, such that

$ax + by \geq c$ for red points

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

The missing slide from last time: Choosing the correct model capacity



Naive Bayes is a linear classifier

- Binary Naive Bayes. We compute:

$$\log \frac{P(C \mid d)}{P(\overline{C} \mid d)} = \log \frac{P(C)}{P(\overline{C})} + \sum_{w \in d} \log \frac{P(w \mid C)}{P(w \mid \overline{C})}$$

- Decide class C if the odds ratio is greater than 1, i.e., if the log odds is greater than 0.
- So decision boundary is hyperplane:

$$\alpha + \sum_{w \in V} \beta_w \times n_w = 0 \quad \text{where } \alpha = \log \frac{P(C)}{P(\overline{C})};$$

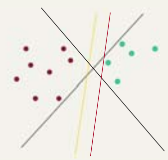
$$\beta_w = \log \frac{P(w \mid C)}{P(w \mid \overline{C})}; \quad n_w = \#\text{ of occurrences of } w \text{ in } d$$

This Week's Topics: More Text Classification

- Today
  - One more machine learning method for text classification
    - Support vector machines
    - Some empirical evaluation and comparison
- Thursday (last class!)
  - Text-specific issues in classification

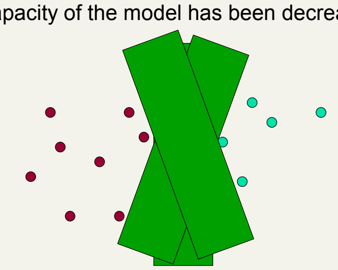
Which Hyperplane?

- 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



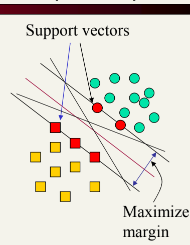
Another intuition

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



Support Vector Machine (SVM)

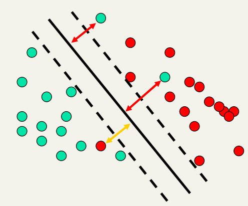
- SVMs maximize the *margin* around the separating hyperplane.
  - A.k.a. large margin classifiers
- The decision function is fully specified by a subset of training samples, *the support vectors*.
- *Quadratic programming* problem
- Seen by many as most successful current text classification method



Large margin classifiers

If **not linearly separable**

- Allow some errors
  - Let some points be moved to where they belong, at a cost
- Still, try to place hyperplane "far" from each class

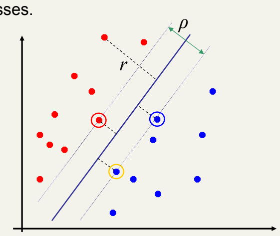


Maximum Margin: Formalization

- w: decision hyperplane normal
- x_i: data point i
- y_i: class of data point i (+1 or -1) NB: Not 1/0
- Classifier is: $\text{sign}(w^T x_i + b)$
- 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….
- Functional margin of dataset is min of above

Geometric Margin

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



Linear SVM Mathematically

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

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

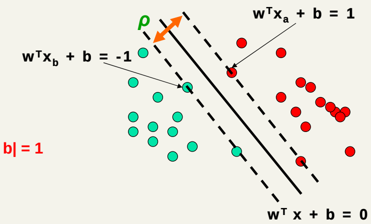
$$\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b \le -1 \quad \text{if } y_i = -1$$

- For support vectors, the inequality becomes an equality; then, since each example's distance from the hyperplane is $r = \frac{\mathbf{w}^T\mathbf{x} + b}{\|\mathbf{w}\|}$ the margin is:

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

Linear Support Vector Machine (SVM)

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



Linear SVMs Mathematically (cont.)

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

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

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

$\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b \ge 1$ if $y_i = 1$; $\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b \le -1$ if $y_i = -1$

- A better formulation (min **||w||** = max 1/ **||w||**):

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

$\mathbf{\Phi}(\mathbf{w}) = ½\, \mathbf{w}^{\mathrm{T}}\mathbf{w}$ is minimized;

and for all $\{(\mathbf{x_i}, y_i)\}$: $y_i\,(\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b) \ge 1$

Solving the Optimization Problem

Find $\mathbf{w}$ and b such that
$\mathbf{\Phi}(\mathbf{w}) = ½\, \mathbf{w}^{\mathrm{T}}\mathbf{w}$ is minimized;
and for all $\{(\mathbf{x_i}, y_i)\}$: $y_i\,(\mathbf{w}^{\mathbf{T}}\mathbf{x_i} + b) \ge 1$

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

Find $\alpha_1 \ldots \alpha_N$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - ½\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j}$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \ge 0$ for all α_i

The Optimization Problem Solution

- The solution has the form:

$\mathbf{w} = \Sigma\alpha_i y_i \mathbf{x_i}$ $\quad b = y_k - \mathbf{w}^{\mathbf{T}}\mathbf{x_k}$ for any $\mathbf{x_k}$ such that $\alpha_k \neq 0$

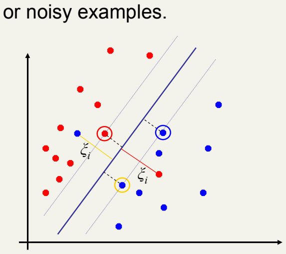
- Each non-zero α_i indicates that corresponding $\mathbf{x_i}$ is a support vector.
- Then the classifying function will have the form:

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

- 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 $\mathbf{x_i}^{\mathbf{T}}\mathbf{x_j}$ between all pairs of training points.

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.



Soft Margin Classification Mathematically

- The old formulation:

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

- The new formulation incorporating slack variables:

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

- Parameter C can be viewed as a way to control overfitting.

Soft Margin Classification – Solution

- The dual problem for soft margin classification:

Find $\alpha_1 \ldots \alpha_N$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x_i}^T\mathbf{x_j}$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $0 \leq \alpha_i \leq C$ for all α_i

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

$\mathbf{w} = \Sigma\alpha_i y_i \mathbf{x_i}$
$b = y_k(1 - \xi_k) - \mathbf{w}^T\mathbf{x_k}$ where $k = \underset{k}{\operatorname{argmax}}\ \alpha_k$

But $\mathbf{w}$ not needed explicitly for classification!

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

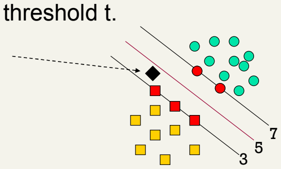
Classification with SVMs

- Given a new point ($x1,x2$), 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



Linear SVMs: Summary

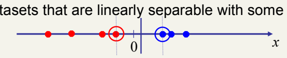
- The classifier is a *separating hyperplane.*
- Most "important" training points are support vectors; they define the hyperplane.
- Quadratic optimization algorithms can identify which training points $\mathbf{x_i}$ are support vectors with non-zero Lagrangian multipliers α_i.
- Both in the dual formulation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

Find $\alpha_1 \ldots \alpha_N$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \boxed{\mathbf{x_i}^T\mathbf{x_j}}$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $0 \leq \alpha_i \leq C$ for all α_i

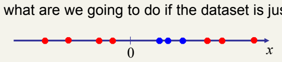
$f(\mathbf{x}) = \Sigma\alpha_i y_i \boxed{\mathbf{x_i}^T\mathbf{x}} + b$

Non-linear SVMs

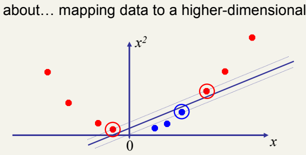
- Datasets that are linearly separable with some noise work out great:



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

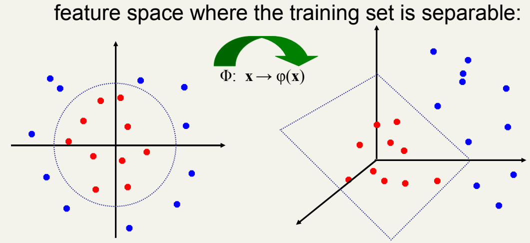


- How about… mapping data to a higher-dimensional space:



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:



The "Kernel Trick"

- The linear classifier relies on 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 Φ: $\mathbf{x} \rightarrow \varphi(\mathbf{x})$, the inner product becomes:

  $$K(\mathbf{x}_i,\mathbf{x}_j)= \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$$
- 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 $K(\mathbf{x}_i,\mathbf{x}_j)= \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$:

  $K(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^T\mathbf{x}_j)^2 = 1+ x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2}+ x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}=$

  $= [1 \;\; x_{i1}^2 \;\; \sqrt{2}\, x_{i1}x_{i2} \;\; x_{i2}^2 \;\; \sqrt{2}x_{i1} \;\; \sqrt{2}x_{i2}]^T [1 \;\; x_{j1}^2 \;\; \sqrt{2}\, x_{j1}x_{j2} \;\; x_{j2}^2 \;\; \sqrt{2}x_{j1} \;\; \sqrt{2}x_{j2}]$

  $= \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$ where $\varphi(\mathbf{x}) = [1 \;\; x_1^2 \;\; \sqrt{2}\, x_1x_2 \;\; x_2^2 \;\; \sqrt{2}x_1 \;\; \sqrt{2}x_2]$

Kernels

- Why use kernels?
  - Make non-separable problem separable.
  - Map data into better representational space
- Common kernels
  - Linear
  - Polynomial $K(x,z) = (1+x^Tz)^d$
  - Radial basis function (infinite space)

$$K(\mathbf{x}_i,\mathbf{x}_j) = e^{-\|\mathbf{x}_i-\mathbf{x}_j\|^2/2\sigma^2}$$

SVMs: Predicting Generalization

- We want the classifier with the best generalization (best accuracy on new data).
- What are clues for good generalization?
  - Large training set
  - Low error on training set
  - Capacity/variance (number of parameters in the model, expressive power of model)
- SVMs give you an explicit bound on error on new data based on these.

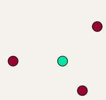
Capacity/Variance: VC Dimension

- Theoretical risk boundary:

  $$R(\alpha) \le R_{emp}(\alpha) + \sqrt{\left(\frac{h(\log(2l/h)+1)-\log(\eta/4)}{l}\right)}$$
- Risk = mean error rate
- α – the model (defined by its parameters)
- R_{emp} - empirical risk, l - #observations, h – VC dimension, the above holds with prob. (1-η)
  - VC (Vapnik-Chervonenkis) dimension/Capacity: max number of points that can be shattered
  - A set can be shattered if the classifier can learn every possible labeling.
- Important theoretical property; Not very often used in practice

Exercise

- Suppose you have n points in d dimensions, labeled red or green. How big need n be (as a function of d) in order to create an example with the red and green points not linearly separable?
- E.g., for $d=2$, $n \ge 4$.



Sketch Theoretical Justification for Maximum Margins

- Vapnik has proved the following:

  The class of optimal linear separators has VC dimension h bounded from above as $h \le \min\left\{\left\lceil \frac{D^2}{\rho^2} \right\rceil, m_0\right\} + 1$

  where ρ is the margin, D is the diameter of the smallest sphere that can enclose all of the training examples, and m_0 is the dimensionality.
- Intuitively, this implies that regardless of dimensionality m_0 we can minimize the VC dimension by maximizing the margin ρ.
- Thus, complexity of the classifier is kept small regardless of dimensionality.

Performance of SVM

- SVM are seen as best-performing method by many.
- Statistical significance of most results not clear.
- There are many methods that perform about as well as SVM.
- Example: regularized logistic regression (Zhang&Oles)
- Example of a comparison study: Yang & Liu

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 (#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)

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.

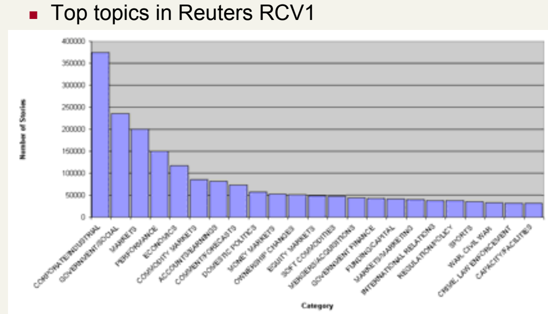
Delegates to the three day Congress will be considering 26 resolutions concerning various issues, including the future direction of farm policy and the tax law as it applies to the agriculture sector. The delegates will also debate whether to endorse concepts of a national PRV (pseudorabies virus) control 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. Reuter

&#3;</BODY></TEXT></REUTERS>

New Reuters: RCV1: 810,000 docs

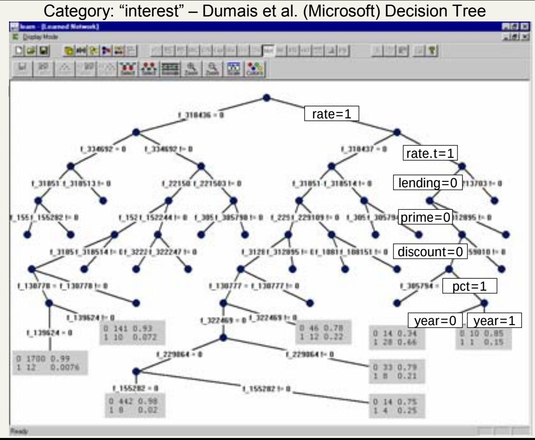
- Top topics in Reuters RCV1



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: $\frac{\sum_i c_{ii}}{\sum_j \sum_i c_{ij}}$

Category: "interest" – Dumais et al. (Microsoft) Decision Tree



Dumais et al. 1998: Reuters - Accuracy

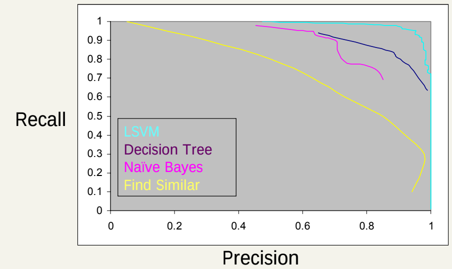
	Rocchio	NBayes	Trees	LinearSVM
earn	92.9%	95.9%	97.8%	98.2%
acq	64.7%	87.8%	89.7%	92.8%
money-fx	46.7%	56.6%	66.2%	74.0%
grain	67.5%	78.8%	85.0%	92.4%
crude	70.1%	79.5%	85.0%	88.3%
trade	65.1%	63.9%	72.5%	73.5%
interest	63.4%	64.9%	67.1%	76.3%
ship	49.2%	85.4%	74.2%	78.0%
wheat	68.9%	69.7%	92.5%	89.7%
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

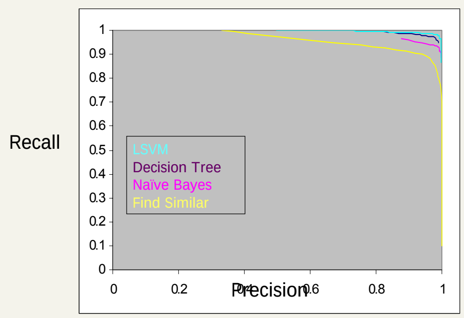
Reuters ROC - Category Grain



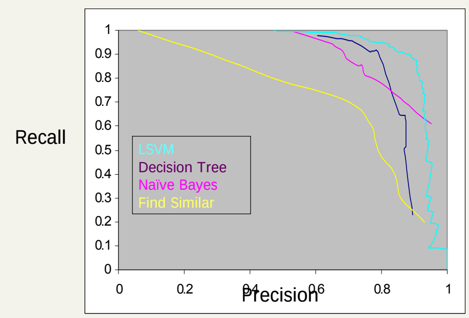
Recall: % labeled in category among those stories that are really in category

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

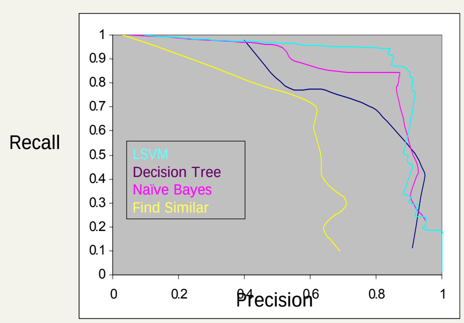
ROC for Category - Earn



ROC for Category - Crude



ROC for Category - Ship



Results for Kernels (Joachims)

	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			

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.

Yang&Liu: Statistical Significance

Table 2: Statistical significance test results

sysA	sysB	s-test	S-test	T-test	T'-test
SVM	kNN	$>$	$\sim$	$\sim$	$\sim$
SVM	LLSF	$\gg$	$\sim$	$\sim$	$\sim$
kNN	LLSF	$\gg$	$\sim$	$\sim$	$\sim$
SVM	NNet	$\gg$	$\gg$	$\gg$	$\gg$
kNN	NNet	$\gg$	$\gg$	$\gg$	$\gg$
LLSF	NNet	$\sim$	$\gg$	$\gg$	$\gg$
NB	kNN	$\ll$	$\ll$	$\ll$	$\ll$
NB	LLSF	$\ll$	$\ll$	$\ll$	$\ll$
NB	SVM	$\ll$	$\ll$	$\ll$	$\ll$
NB	NNet	$\ll$	$\sim$	$\sim$	$\sim$

"$\gg$" or "$\ll$" means P-value ≤ 0.01;
"$>$" or "$<$" means $0.01 <$ P-value ≤ 0.05;
"$\sim$" means P-value > 0.05.

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
  - Bound on "risk" (expected error on test set)
  - Best performing text classifier?
  - Partly popular due to availability of SVMlight
    - SVMlight is accurate and fast – and free (for research)
  - Now lots of software: libsvm, TinySVM, ….

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. *Proceedings of 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.