Appendix 1 to Lecture 18 of "Machine Learning"

Inductive Learning Algorithms for Text Classification

Susan Dumais
David Heckerman
Eric Horvitz
John Platt
(Microsoft Research)
Mehran Sahami
(Stanford Univ)

Berkeley March 13, 1998

Road Map

- Text Classification Basics
- Inductive Learning Methods
- Reuters Results
- Future Plans

Text Classification - Basics

- *Text Classification* - put objects into groups, using textual descriptions
- Examples:
  - *Text Categorization* - assign documents to one or more of a predefined set of categories
  - *Text Retrieval* - distinguish items that are relevant/non_relevant to user's query
  - *Text Discovery* - discovery of groupings (e.g., clusters) and other patterns in data

Text Categorization - Applications

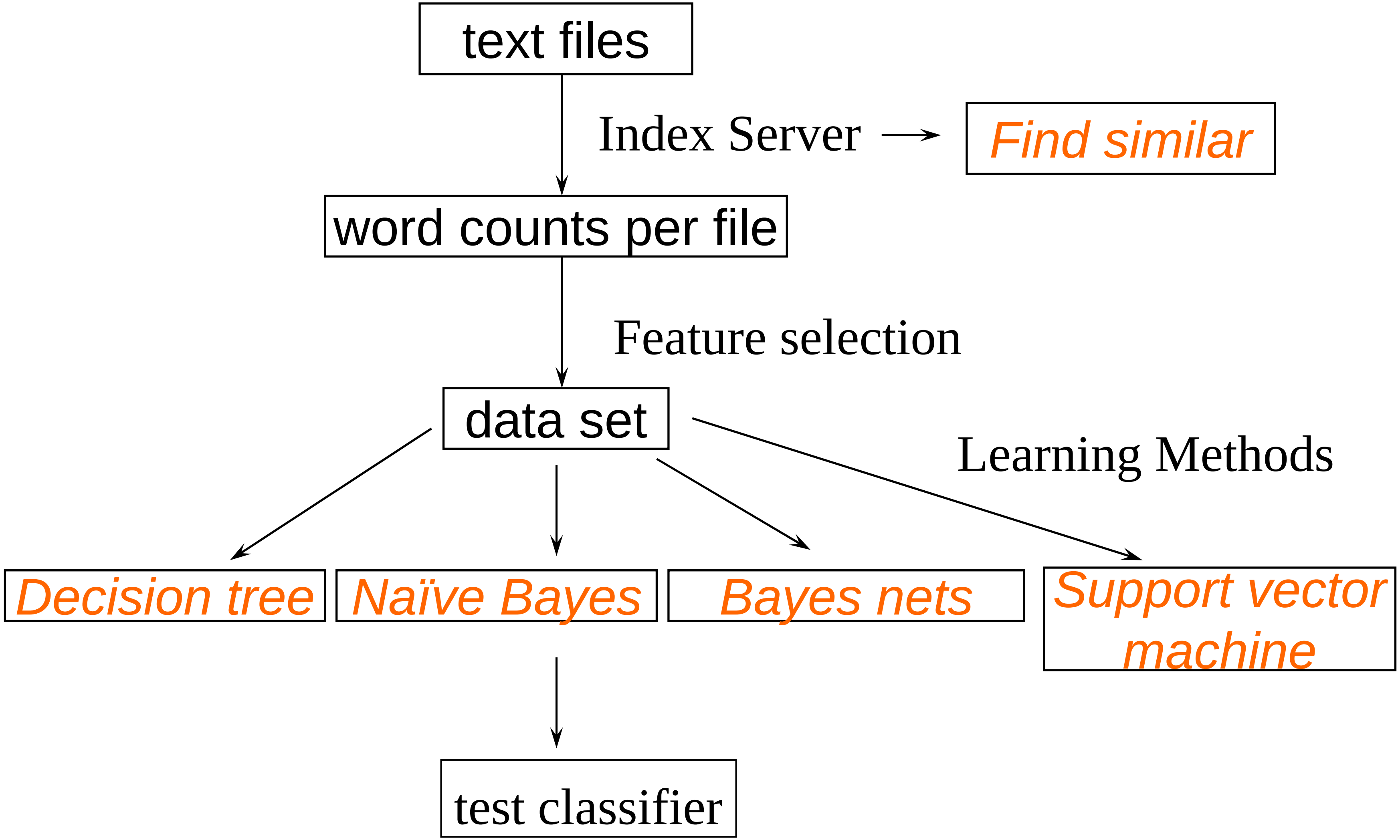
- Sorting new items into existing structures (e.g., email folders, general file system, site ontologies, objectionable vs. not)
- Routing user's requests
- Topic specific processing
- Structured browsing & search
- Information filtering/push
- Dynamic interests

Text Classification - Methods

- Human classifiers (e.g., Dewey, LC, MeSH, Yahoo!, CyberPatrol)
- Hand-crafted knowledge engineered systems (e.g., CONSTRUE)

- Inductive learning methods
  - (Semi-) automatic classification

Text Classification Process



Learning Methods

- A classifier is a function: $f(\mathbf{x}) = p(class)$
  - *from* attribute vectors, $\mathbf{x}=(x_1,x_2, \ldots x_d)$
  - *to* target values, p(class)
- Example classifiers
  - (interest AND rate) OR (quarterly) -> "interest"
  - score = 0.3*interest + 0.4*rate + 0.1*quarterly; if score > .8, then "interest" category

Inductive Learning Methods

- Supervised learning to build classifiers
  - Labeled training data (i.e., examples of items in each category)
  - “Learn” classifier
  - Test effectiveness on new instances
- Statistical guarantees of effectiveness

Inductive Learning Methods

- Classifiers easy to construct and update
- Requires only subject knowledge (“I know it when I see it”)
- Customizable for individual’s categories and tasks
- Graded estimates of category membership allow for tradeoffs between precision and recall, depending on task

Text Representation

- Vector space representation of documents

	word1	word2	word3	word4 …
Doc 1 =	<1,	0,	3,	0, ... >
Doc 2 =	<0,	1,	0,	0, ... >
Doc 3 =	<0,	0,	0,	5, ... >

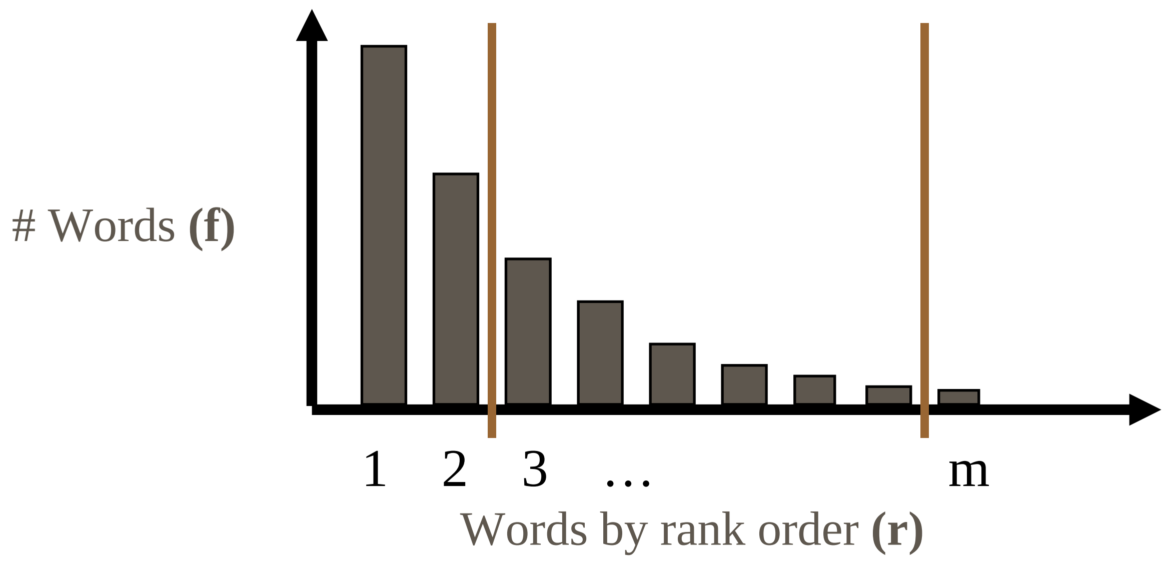
- Mostly use: simple words, binary weights

- Text can have 10^7 or more dimensions

  e.g., 100k web pages had 2.5 million distinct words

Feature Selection

⌘ *Word distribution* - remove frequent and infrequent words based on Zipf's law: **f**requency * **r**ank ~ constant



Feature Selection (cont'd)

- *Fit to categories* - use mutual information to select features which best discriminate category vs. not $MI(x,C) = \sum p(x,C) \log(\frac{p(x,C)}{p(x)p(C)})$
- *Designer features* - domain specific, including non-text features

- Use 100-500 best features from this process as input to learning methods

Inductive Learning Methods

- Find Similar
- Decision Trees
- Naïve Bayes
- Bayes Nets
- Support Vector Machines (SVMs)

- All support:
  - "Probabilities" - graded membership; comparability across categories
  - Adaptive - over time; across individuals

Find Similar

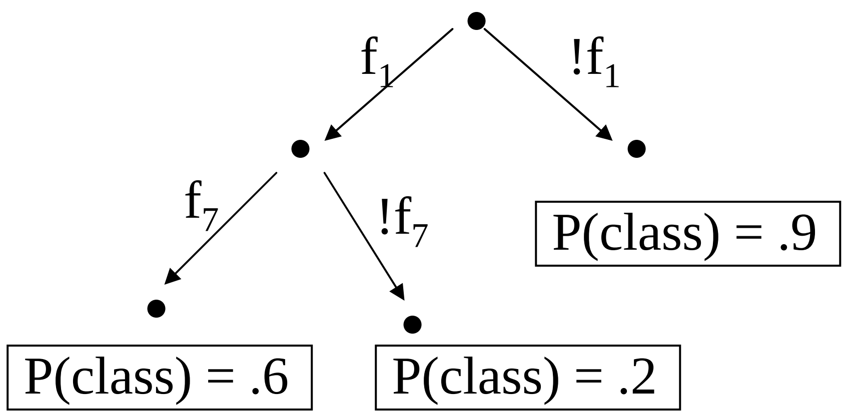
- Aka, relevance feedback
- Rocchio

$$w_j = \beta \sum_{i \notin rel} \frac{x_{i,j}}{n} - \gamma \sum_{i \notin non\_rel} \frac{x_{i,j}}{N-n}$$

- Classifier parameters are a weighted combination of weights in positive and negative examples -- “centroid”
- New items classified using: $\sum_j w_j \cdot x_j$
- Use all features, idf weights, $\gamma = 0$

Decision Trees

- Learn a sequence of tests on features, typically using top-down, greedy search
- Binary (yes/no) or continuous decisions

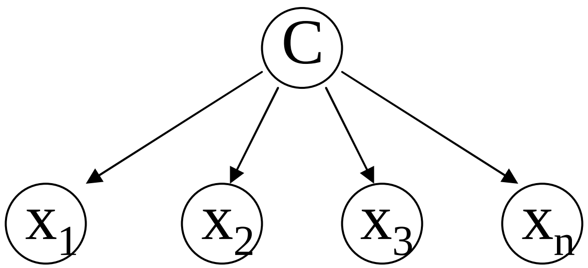


Naïve Bayes

- Aka, binary independence model
- Maximize: Pr (Class | Features)

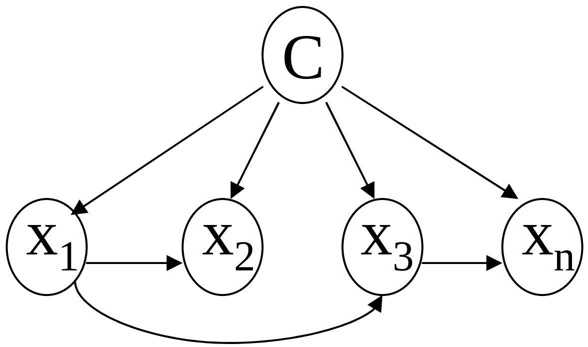
$$P(class \mid \vec{x}) = \frac{P(\vec{x} \mid class) \cdot P(class)}{P(\vec{x})}$$

- Assume features are conditionally independent - math easy; surprisingly effective



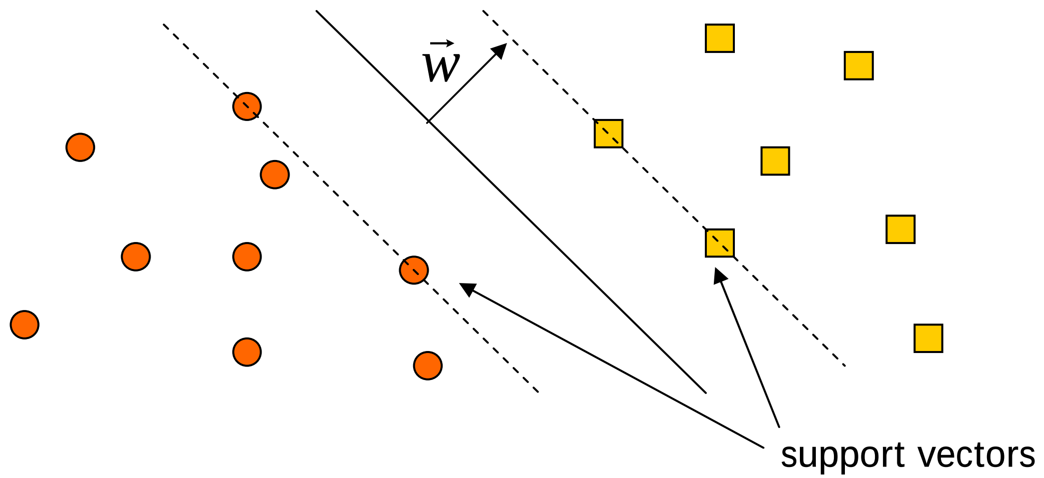
Bayes Nets

- Maximize: Pr (Class | Features)
- Does not assume independence of features - dependency modeling



Support Vector Machines

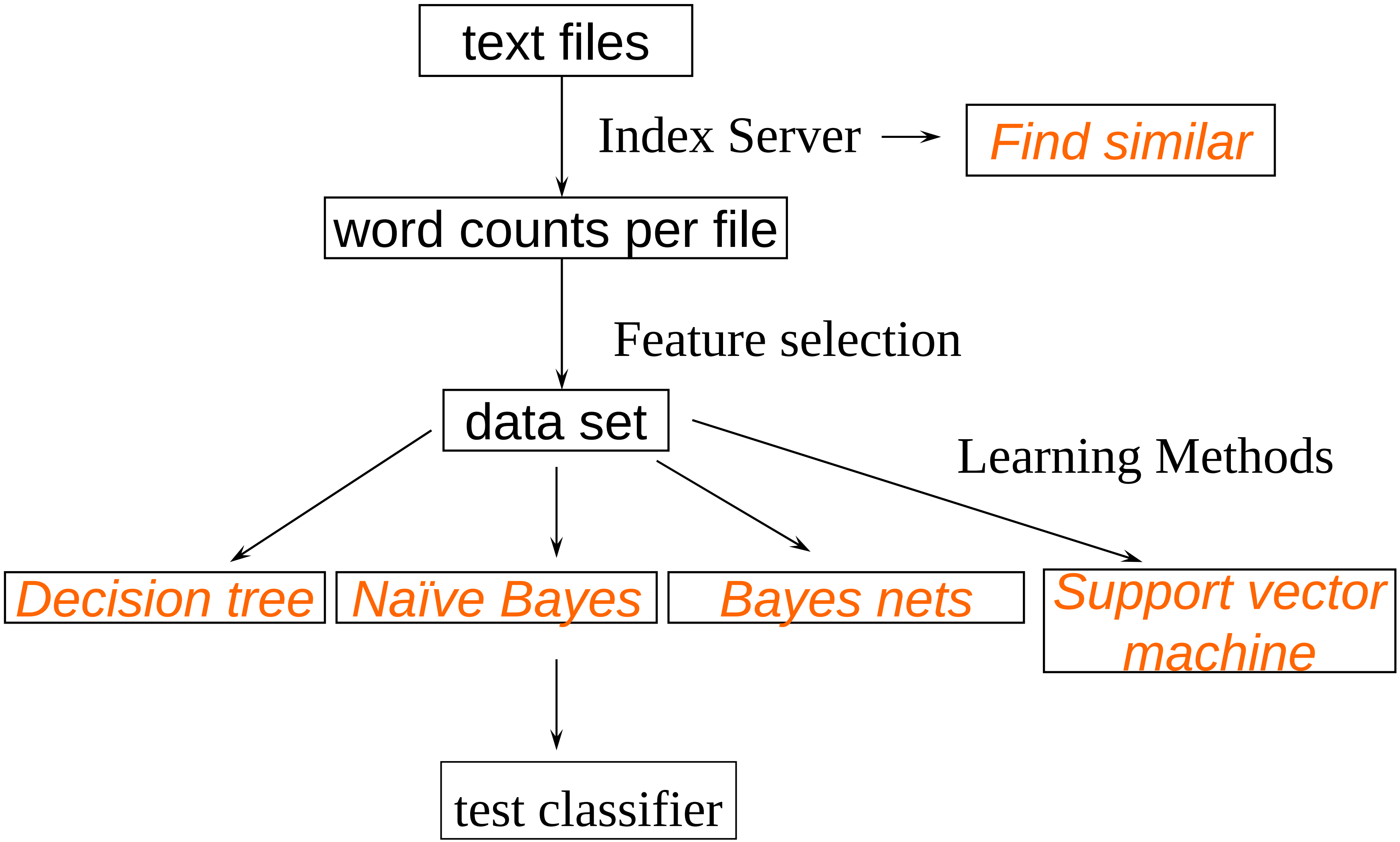
- Vapnik (1979)
- Binary classifiers that maximize margin
  - Find hyperplane separating positive and negative examples
  - Optimization for maximum margin: $\min \|\vec{w}\|^2, \vec{w} \cdot \vec{x} - b \geq 1, \vec{w} \cdot \vec{x} - b \leq -1$
  - Classify new items using: $\vec{w} \cdot \vec{x}$



Support Vector Machines

- Extendable to:
  - Non-separable problems (Cortes & Vapnik, 1995)
  - Non-linear classifiers (Boser et al., 1992)
- Good generalization performance
  - OCR (Boser et al.)
  - Vision (Poggio et al.)
  - Text classification (Joachims)

Text Classification Process



Reuters Data Set
(21578 - ModApte split)

- 9603 training articles; 3299 test articles
- Example "interest" article

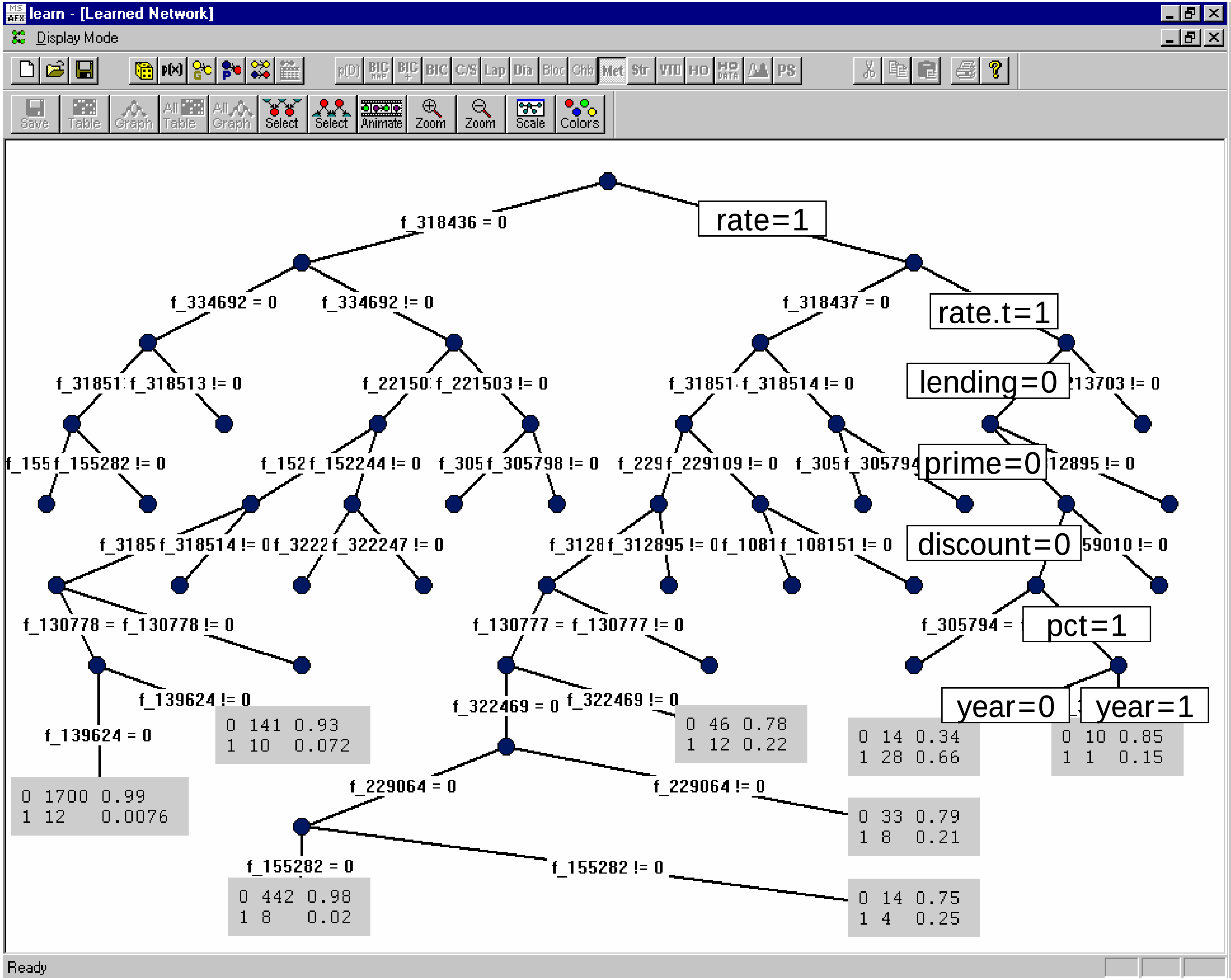
  2-APR-1987 06:35:19.50
  west-germany
  b f BC-BUNDESBANK-LEAVES-CRE 04-02 0052
  FRANKFURT, March 2
  The Bundesbank left credit policies unchanged after today's regular meeting of its council, a spokesman said in answer to enquiries. The West German discount rate remains at 3.0 pct, and the Lombard emergency financing rate at 5.0 pct.
  REUTER

- Average article 200 words long

Reuters Data Set
(21578 - ModApte split)

- 118 categories
  - An article can be in more than one category
  - Learn 118 binary category distinctions
- Most 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)

Category: “interest”



Category: Interest

- Example SVM features - $\vec{w}$
  - 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

Accuracy Scores

- Based on contingency table

	Truth: Yes	Truth: No
System: Yes	*a*	*b*
System: No	*c*	*d*

- Effectiveness measure for binary classification
  - error rate = (b+c)/n
  - accuracy = 1 - error rate
  - precision (P) = a/(a+b)
  - recall (R) = a/(a+c)
  - break-even = (P+R)/2
  - F measure = 2PR/(P+R)

Reuters - Accuracy ((R+P)/2)

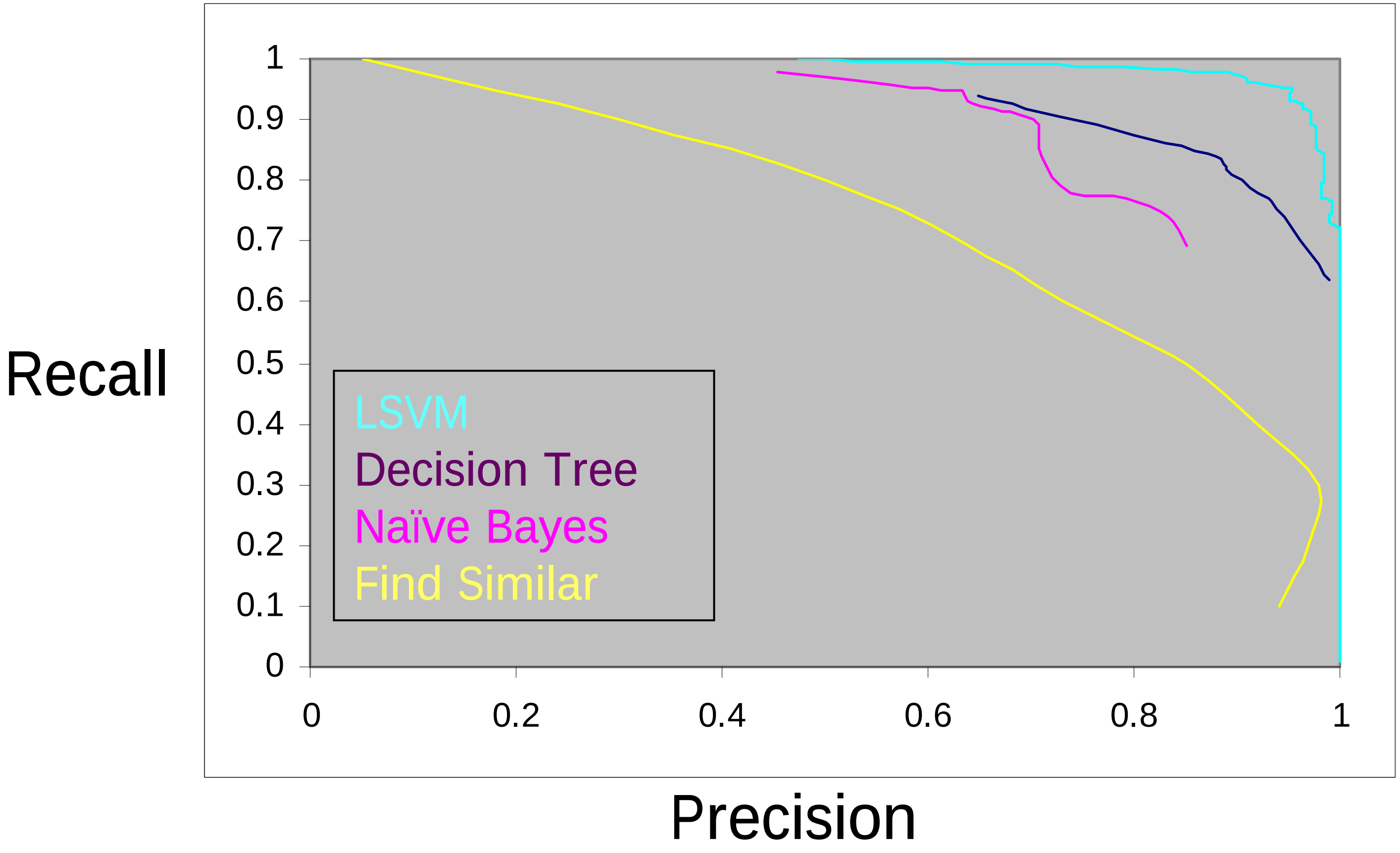
	Findsim	NBayes	BayesNets	Trees	LinearSVM
earn	92.9%	95.9%	95.8%	97.8%	98.2%
acq	64.7%	87.8%	88.3%	89.7%	92.8%
money-fx	46.7%	56.6%	58.8%	66.2%	74.0%
grain	67.5%	78.8%	81.4%	85.0%	92.4%
crude	70.1%	79.5%	79.6%	85.0%	88.3%
trade	65.1%	63.9%	69.0%	72.5%	73.5%
interest	63.4%	64.9%	71.3%	67.1%	76.3%
ship	49.2%	85.4%	84.4%	74.2%	78.0%
wheat	68.9%	69.7%	82.7%	92.5%	89.7%
corn	48.2%	65.3%	76.4%	91.8%	91.1%
Avg Top 10	64.6%	81.5%	85.0%	88.4%	91.4%
Avg All Cat	61.7%	75.2%	80.0%	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

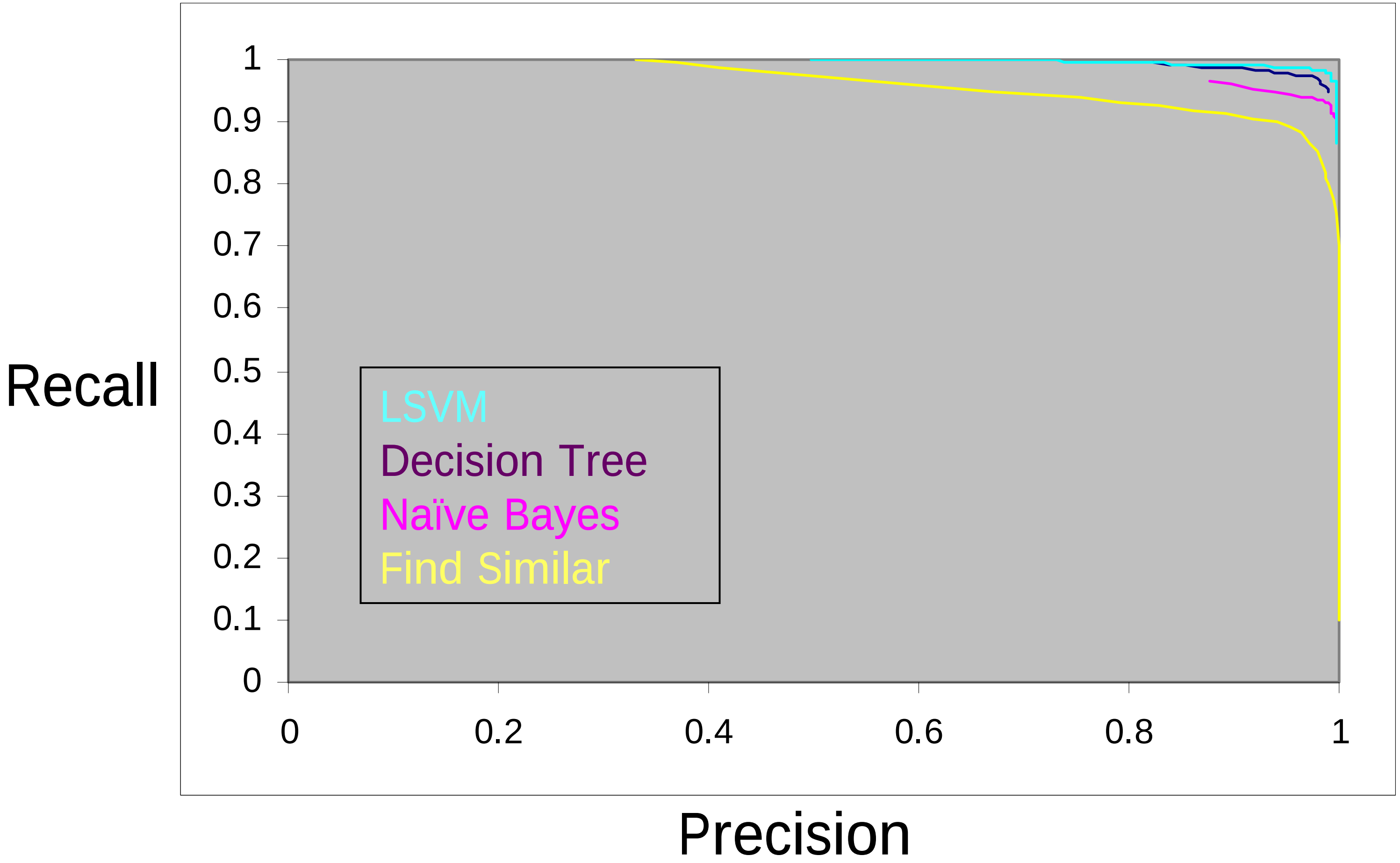
Reuters ROC - Category Grain



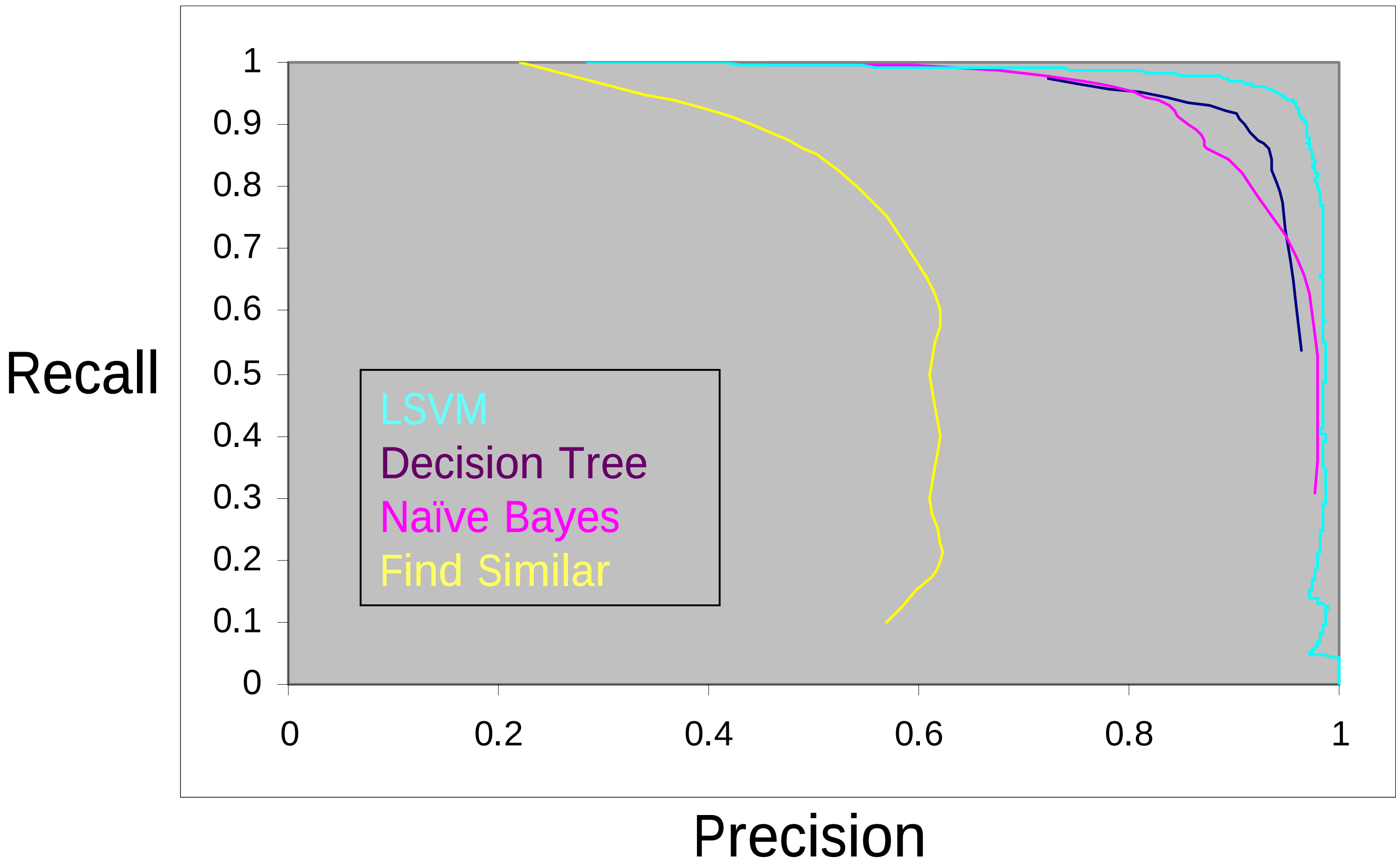
Recall: % labeled in category among those stories that are really in category

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

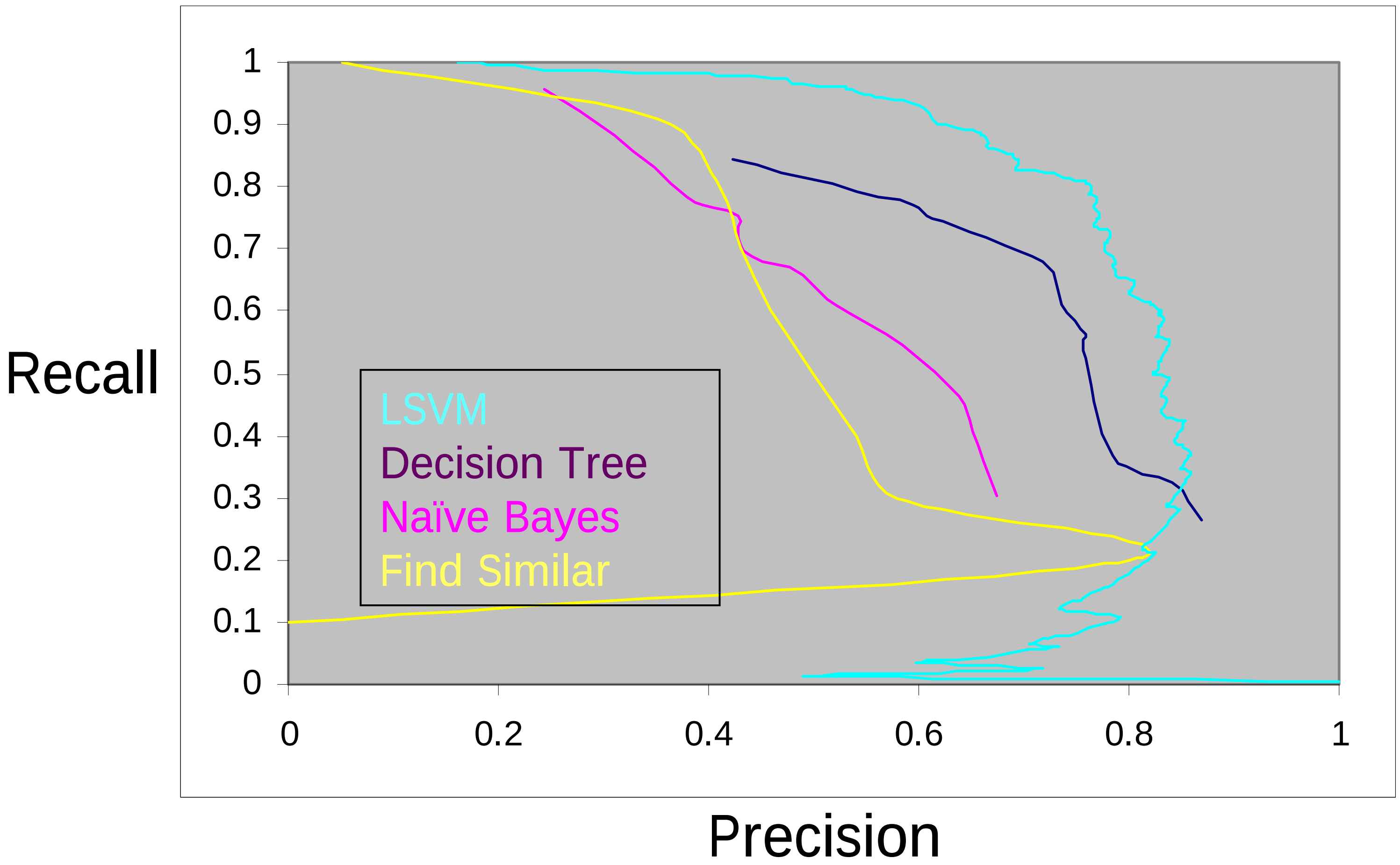
ROC for Category - Earn



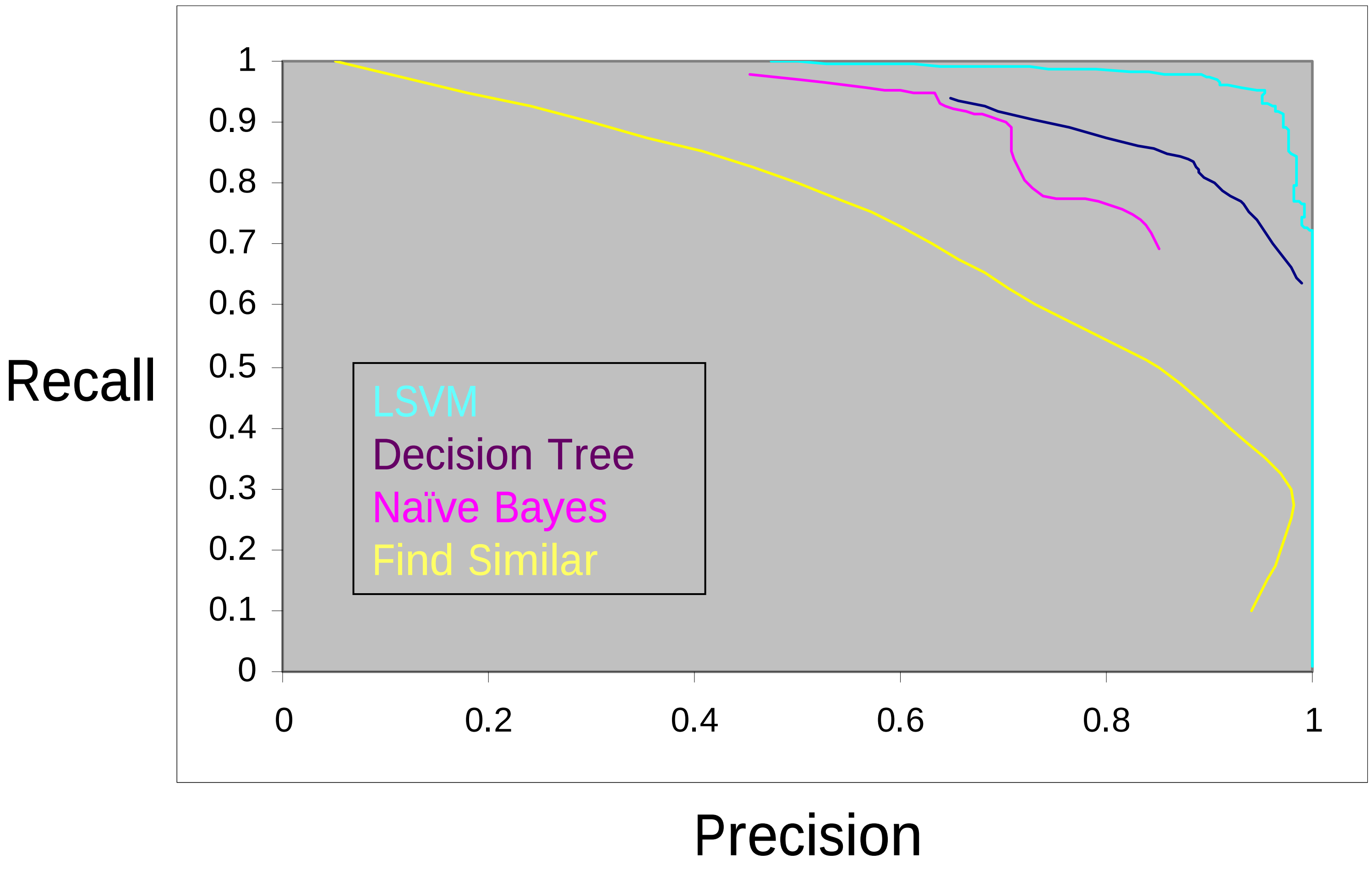
ROC for Category - Acquisitions



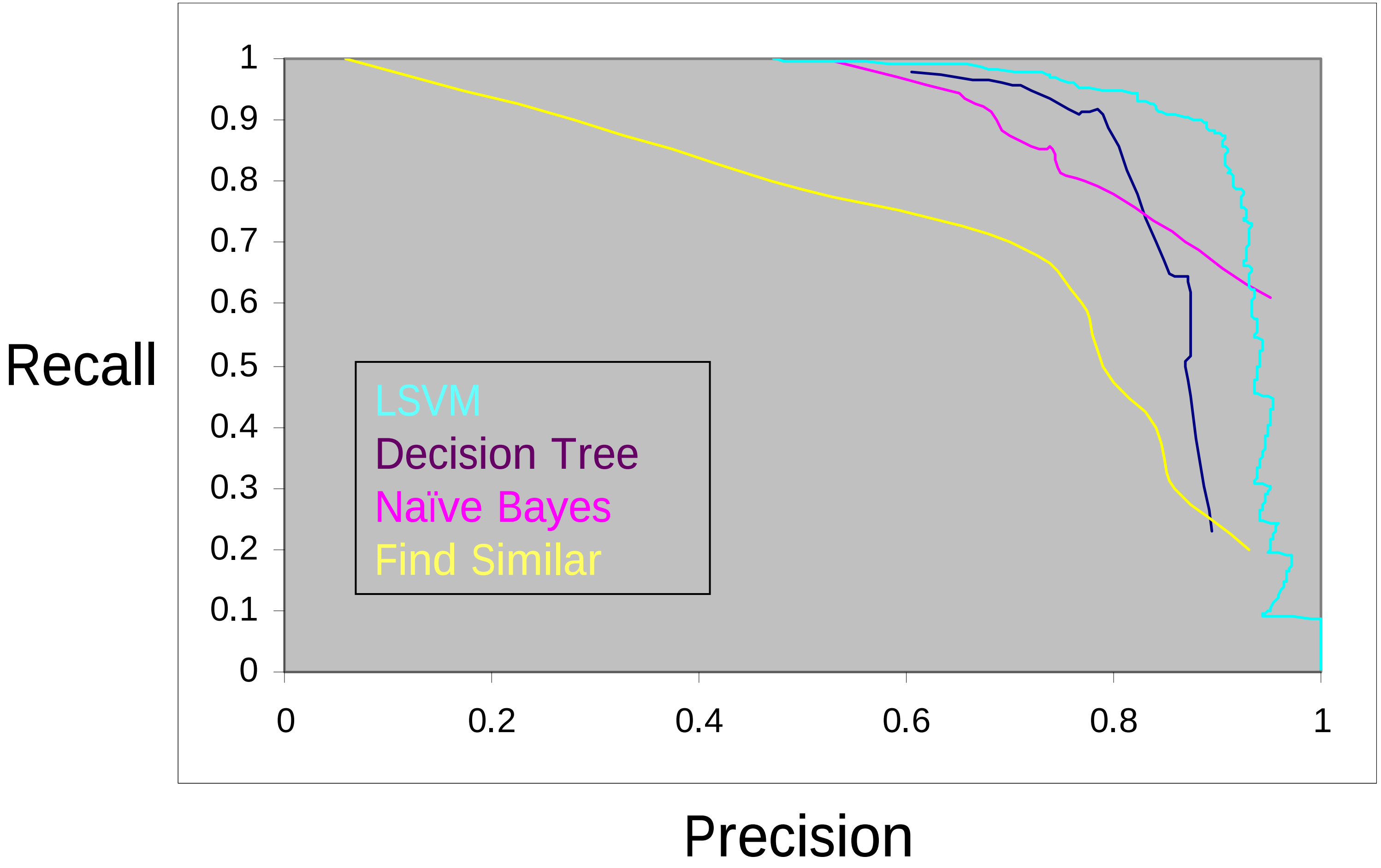
ROC for Category - Money-Fx



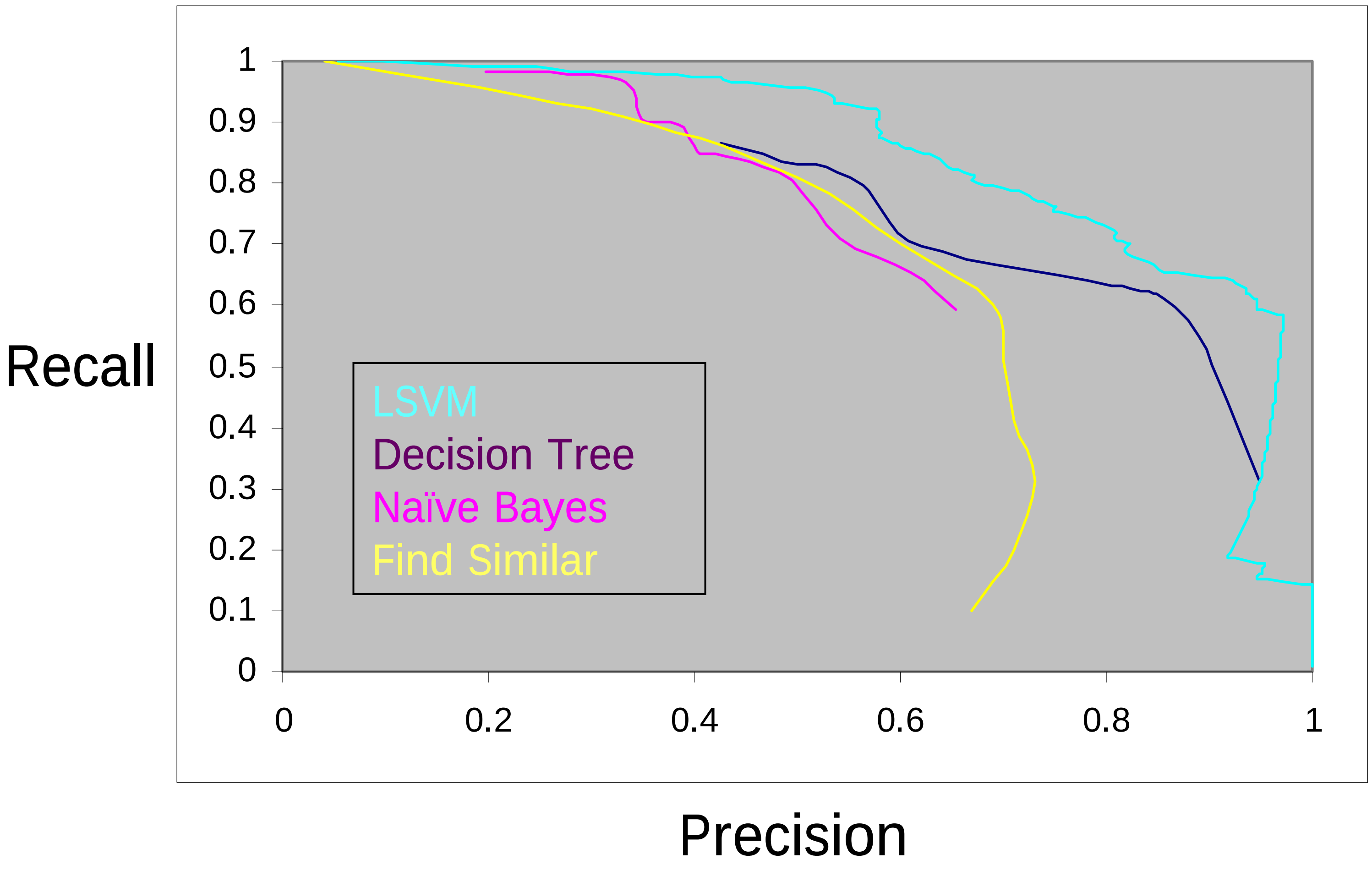
ROC for Category - Grain



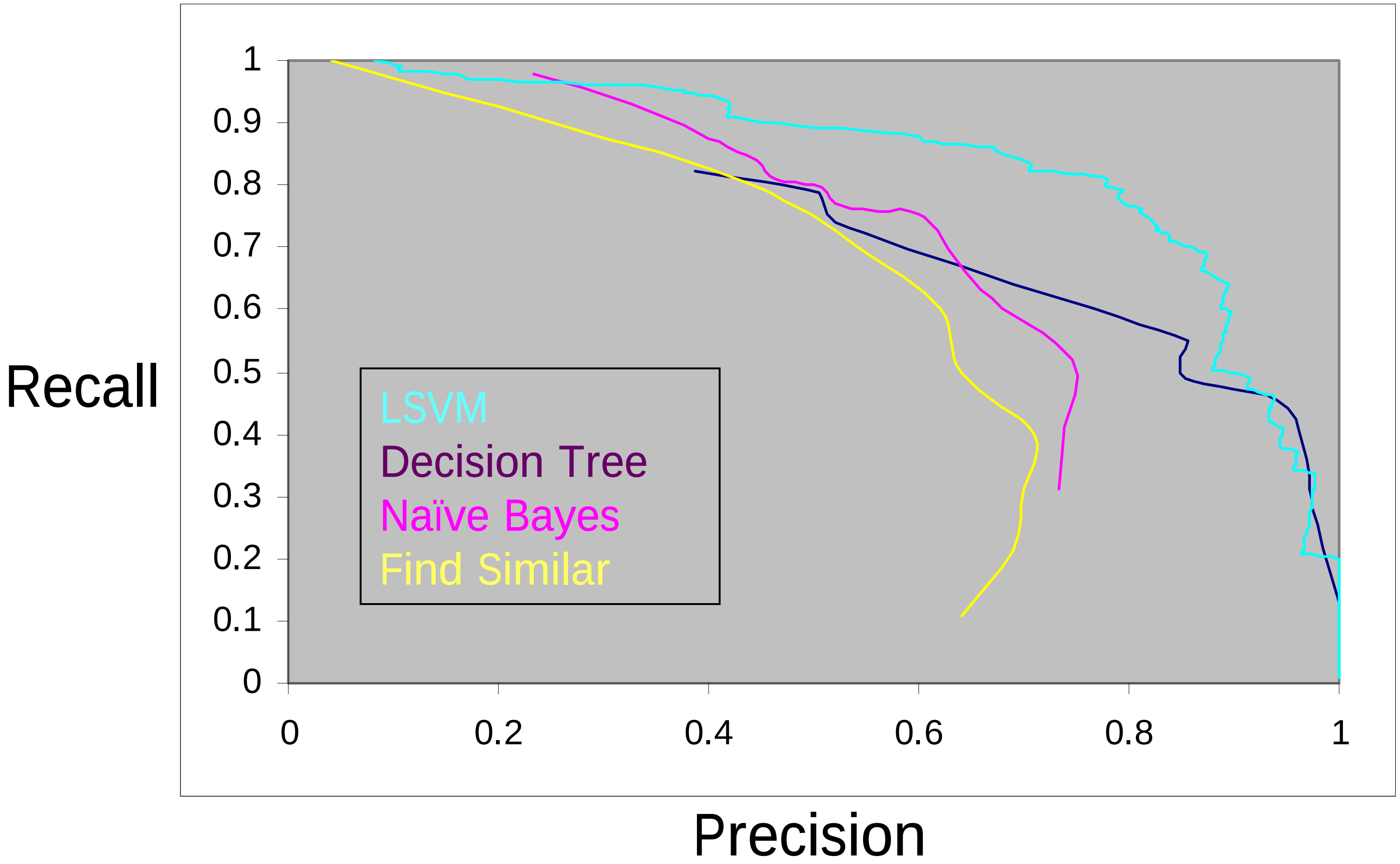
ROC for Category - Crude



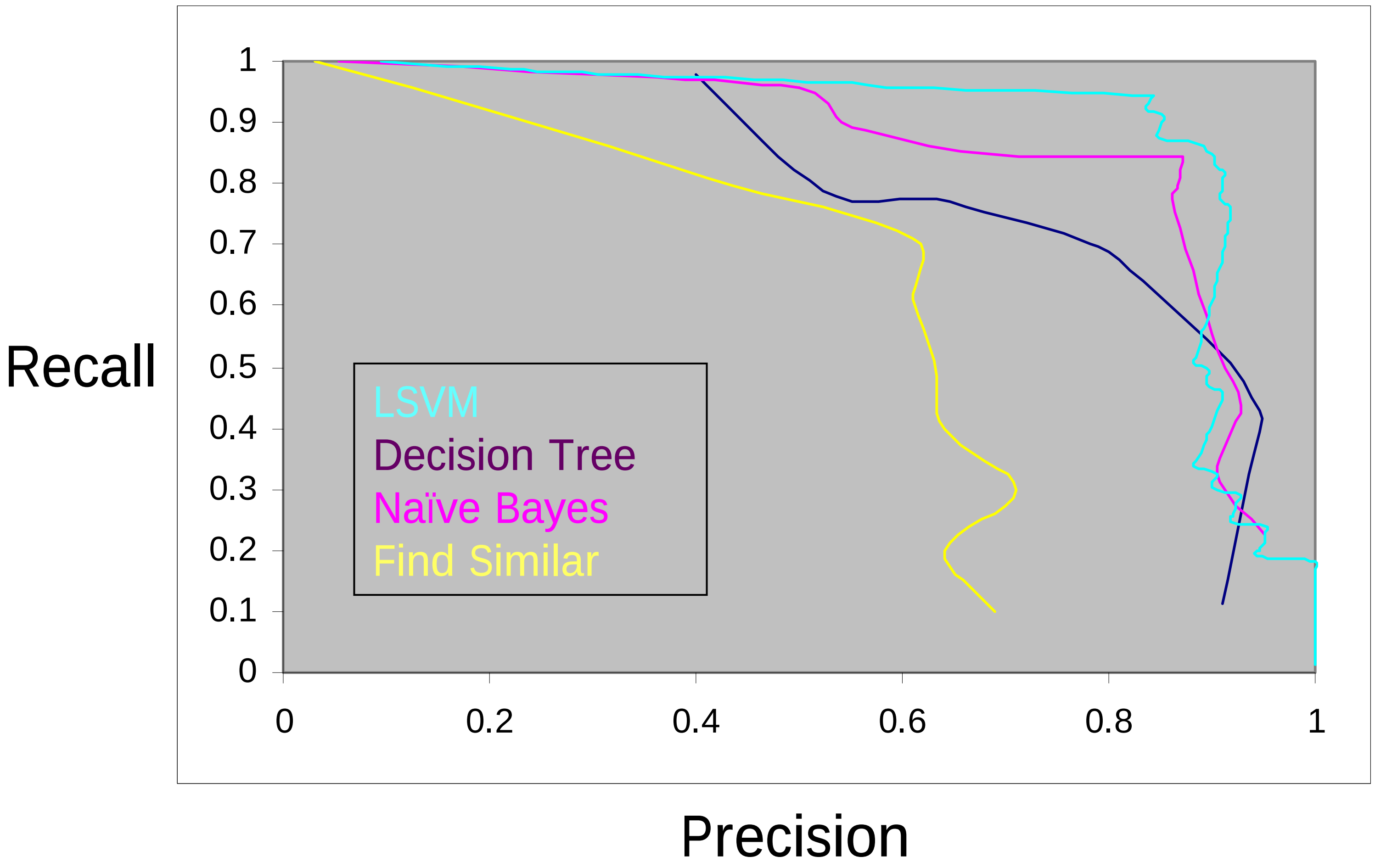
ROC for Category - Trade



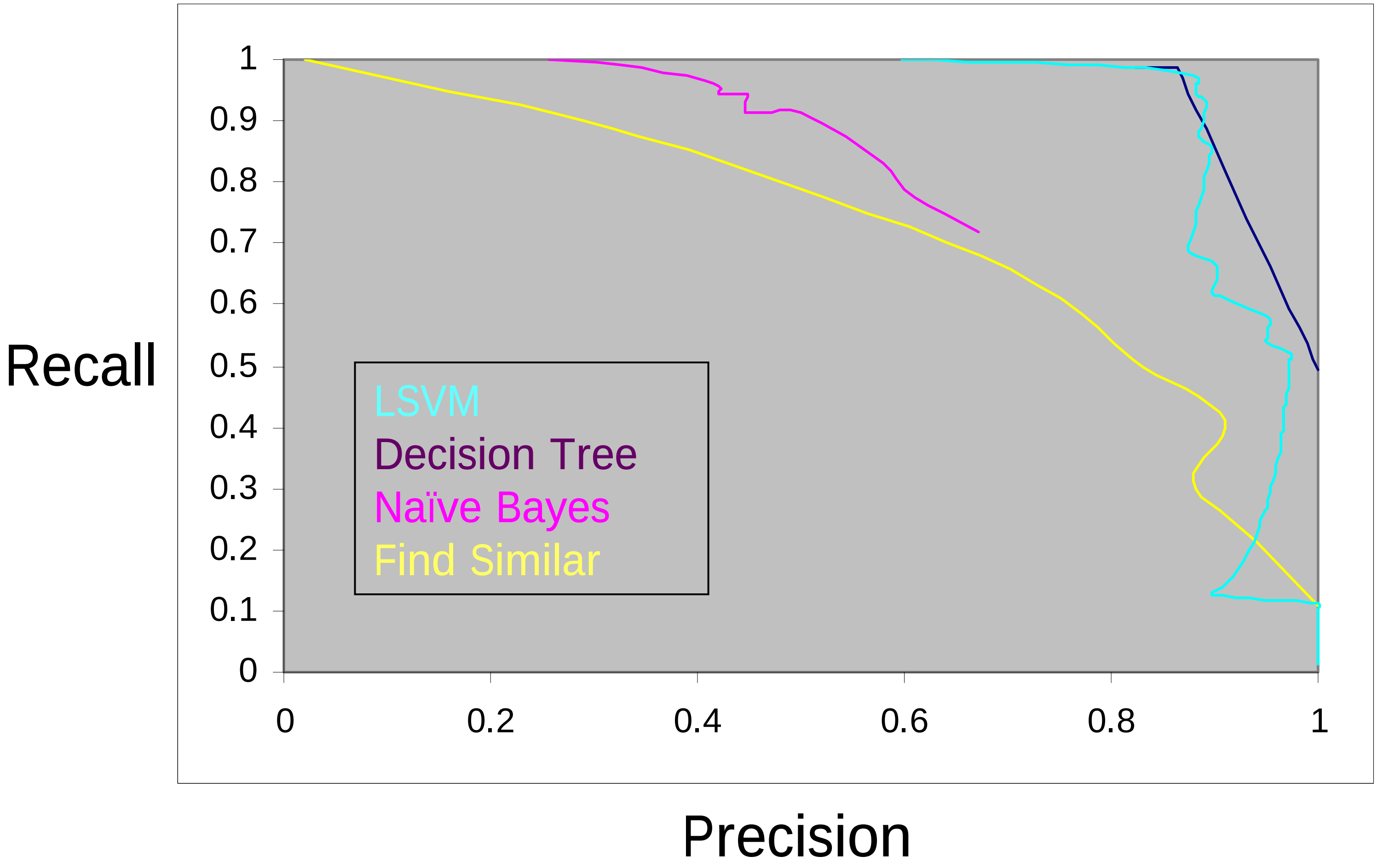
ROC for Category - Interest



ROC for Category - Ship

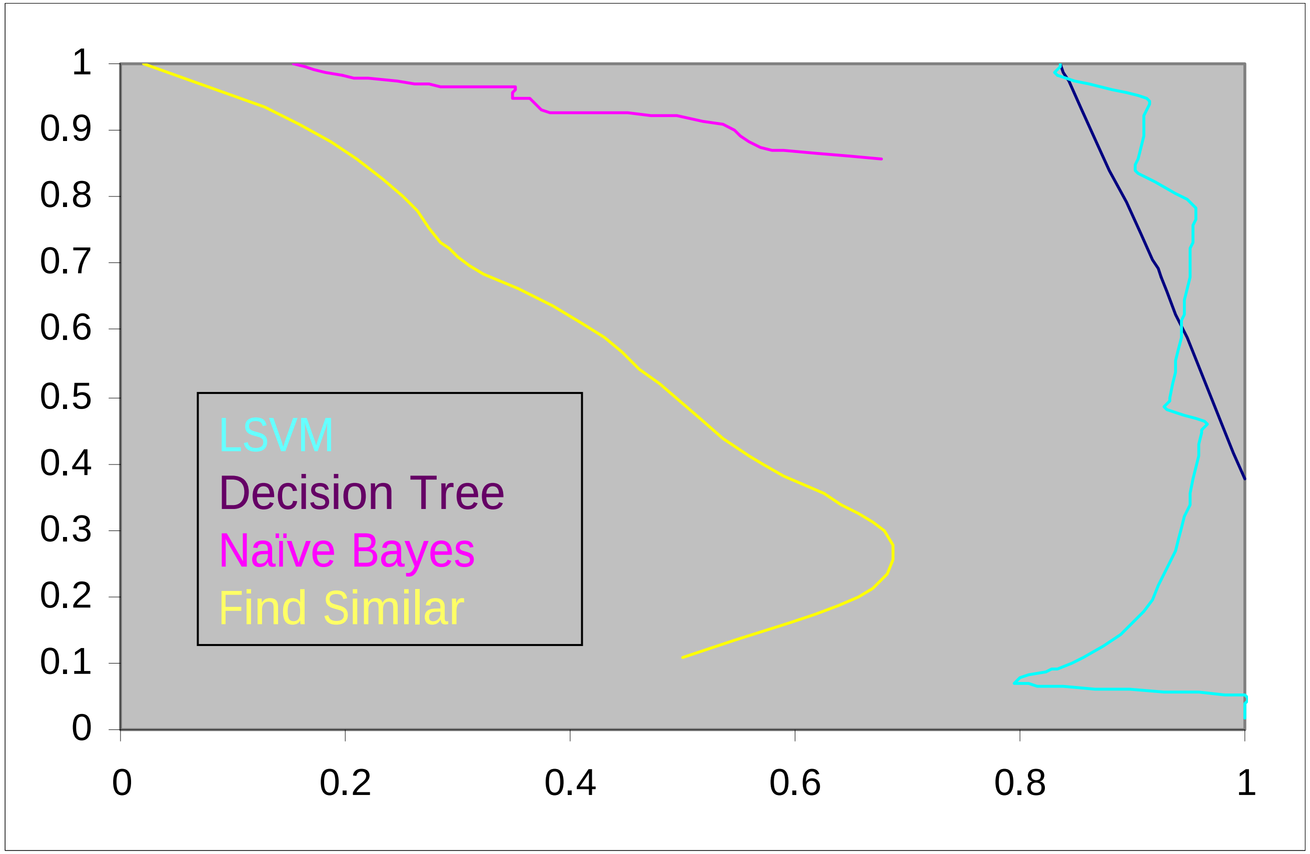


ROC for Category - Wheat



ROC for Category - Corn

Recall



Precision

Reuters - Sample Size (SVM)

sample set 1

category	100% samp sz	(p+r)/2	10% samp sz	(p+r)/2	5% samp sz	(p+r)/2	1% samp sz	(p+r)/2
0-acq	2876	98.3%	281	97.8%	145	97.4%	35	93.6%
1-earn	1650	97.0%	162	94.6%	80	90.4%	14	65.6%
3-money-fx	538	80.2%	55	66.3%	28	63.9%	3	41.9%
4-grain	433	95.9%	46	91.5%	21	87.0%	3	50.3%
5-crude	389	90.4%	45	82.9%	18	76.9%	3	??
6-trade	369	80.9%	40	78.2%	21	76.4%	2	12.0%
7-interest	347	79.9%	32	68.4%	17	55.3%	2	50.8%
8-ship	197	85.5%	20	57.4%	11	53.9%	2	??
9-wheat	212	92.5%	24	84.8%	11	65.7%	2	50.7%
10-corn	182	93.0%	23	78.2%	9	60.3%	1	50.9%
microtop10		93.9%		89.7%		86.0%		70.3%

sample set 2

category	100% samp sz	(p+r)/2	10% samp sz	(p+r)/2	5% samp sz	(p+r)/2	1% samp sz	(p+r)/2
0-acq	2876	98.3%	264	97.6%	139	97.3%	26	94.6%
1-earn	1650	97.0%	184	94.1%	79	90.6%	16	73.5%
3-money-fx	538	80.2%	62	72.9%	29	71.0%	5	49.9%
4-grain	433	95.9%	40	85.8%	28	88.6%	7	76.5%
5-crude	389	90.4%	35	81.6%	15	65.6%	2	??
6-trade	369	80.9%	41	80.1%	22	72.7%	5	51.6%
7-interest	347	79.9%	30	71.8%	15	63.0%	3	45.1%
8-ship	197	85.5%	21	62.8%	10	54.5%	2	50.6%
9-wheat	212	92.5%	19	80.1%	15	80.1%	5	68.3%
10-corn	182	93.0%	16	76.6%	11	69.6%	4	43.4%
microtop10		93.9%		89.6%		86.4%		75.5%

Reuters - Other Experiments

- Simple words vs. NLP-derived phrases
  - NLP-derived phrases
    - factoids (April_8, Salomon_Brothers_International)
    - mulit-word dictionary entries (New_York, interest_rate)
    - noun phrases (first_quarter, modest_growth)
  - No advantage for Find Similar, Naïve Bayes
  - Need to try w/ SVM
- Binary vs. 0/1/2 features
  - No advantage of 0/1/2 for Decision Trees
  - Need to try w/ SVM

Reuters Summary

- Accurate classifiers can be learned automatically from training examples
- Linear SVMs provide very good classification accuracy
  - Better than best previously reported results for this test collection
- Widely applicable, flexible, and adaptable representations

Text Classification Horizon

- Text representation enhancements for SVM model
- Use of hierarchical category structure
- Dynamic interests
- A range of applications
- UI for (semi-) automatic classification