CS276B

Text Information Retrieval, Mining, and Exploitation

Lecture 8

Text Classification III

Feb 11, 2003

(includes slides borrowed from David Blei, Susan Dumais, Andrew McCallum, and Ray Mooney)

But first … a bit more HMM Information Extraction

References

Leslie Pack Kaelbling, Michael L. Littman and Andrew W. Moore. Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research, pages 237-285, May 1996.

Headers

Journal of Artificial Intelligence Research 4 (1996) 237-285 Submitted 9/95; published 5/96

Reinforcement Learning: A Survey

Leslie Pack Kaelbling LPK@CS.BROWN.EDU
Michael L. Littman MLITTMAN@CS.BROWN.EDU
Computer Science Department, Box 1910, Brown University
Providence, RI 02912-1910 USA

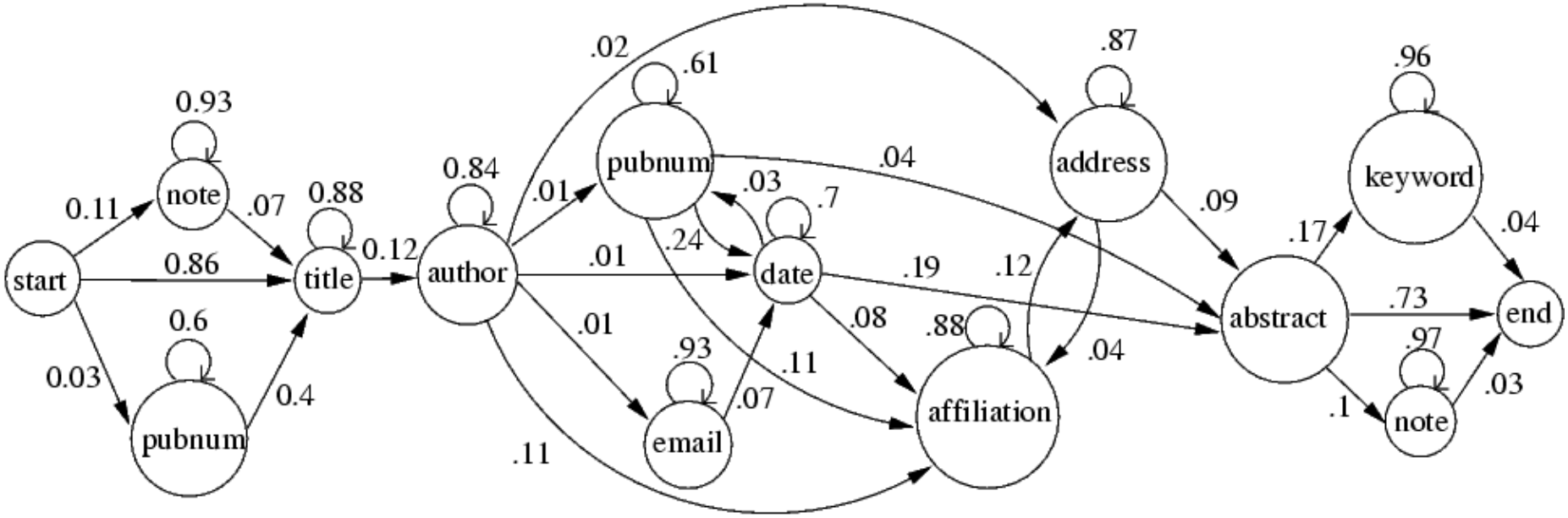
Andrew W. Moore AWM@CS.CMU.EDU
Smith Hall 221, Carnegie Mellon University, 5000 Forbes Avenue
Pittsburgh, PA 15213 USA

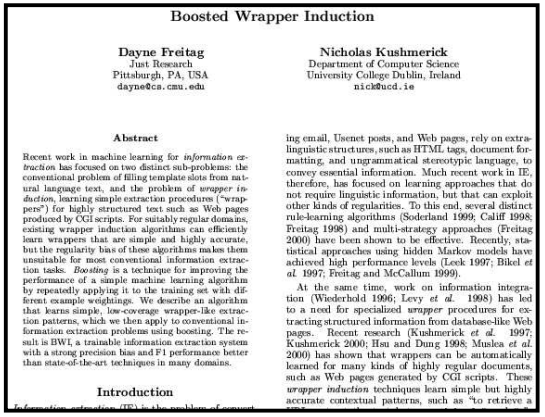
Abstract

This paper surveys the field of reinforcement learning from a computer-science perspective. It is written to be accessible to researchers familiar with machine learning. Both the historical basis of the field and a broad selection of current work are summarized. Reinforcement learning is the problem faced by an agent that learns behavior through trial-and-error interactions with a dynamic environment. The work described here has a resemblance to work in psychology, but differs considerably in the details and in the use of the word "reinforcement." The paper discusses central issues of reinforcement learning, including trading off exploration and exploitation, establishing the foundations of the field via Markov decision theory, learning from delayed reinforcement, constructing empirical models to accelerate learning, making use of generalization and hierarchy, and coping with hidden state. It concludes with a survey of some implemented systems and an assessment of the practical utility of current methods for reinforcement learning.

1 Introduction

HMM for research papers: transitions [Seymore *et al.*, 99]





Boosted Wrapper Induction

Dayne Freitag
Just Research
Pittsburgh, PA, USA
dayne@cs.cmu.edu

Nicholas Kushmerick
Department of Computer Science
University College Dublin, Ireland
nick@ucd.ie

Abstract

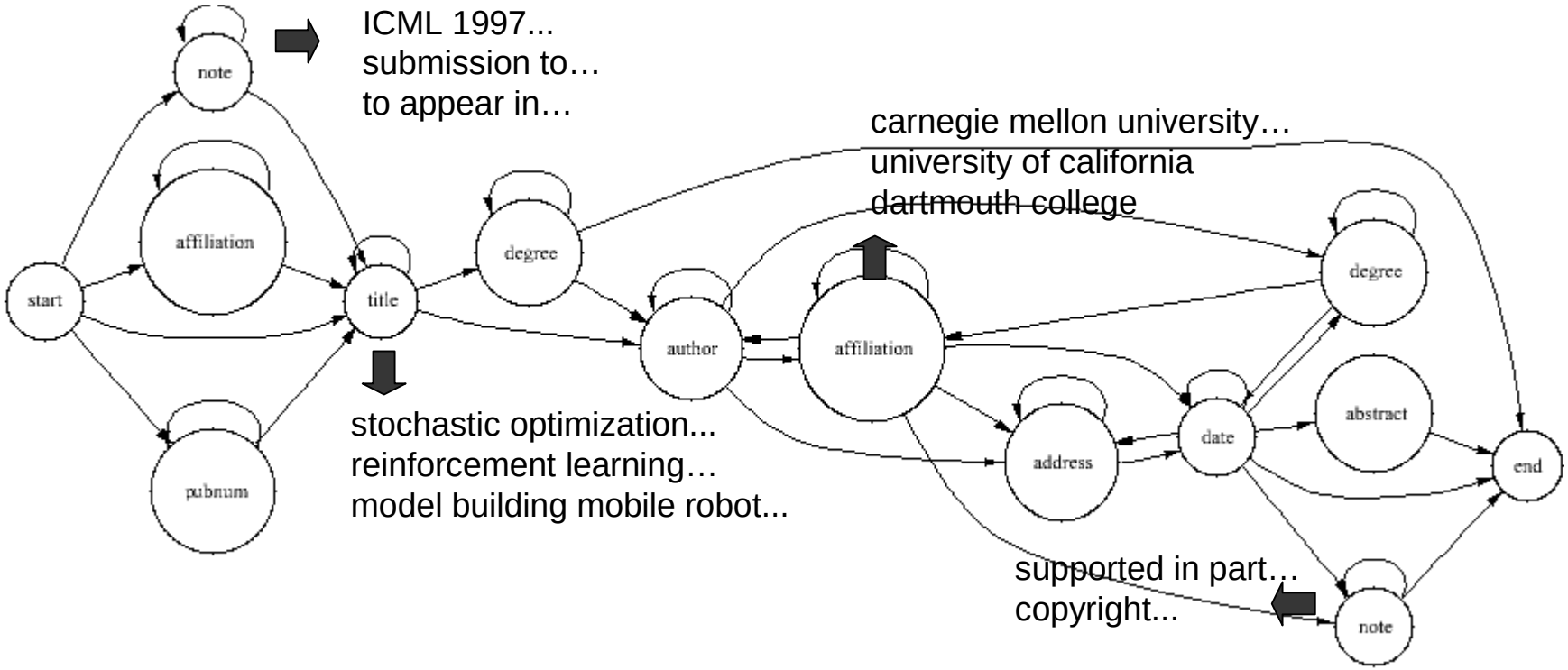
Recent work in machine learning for *information extraction* has focused on two distinct sub-problems: the conventional problem of filling template slots from natural language text, and the problem of *wrapper induction*, learning simple extraction procedures ("wrappers") for highly structured text such as Web pages produced by CGI scripts. For suitably regular domains, existing wrapper induction algorithms can efficiently learn wrappers that are simple and highly accurate, but the regularity bias of these algorithms makes them unsuitable for most conventional information extraction tasks. *Boosting* is a technique for improving the performance of a simple machine learning algorithm by repeatedly applying it to the training set with different example weightings. We describe an algorithm that learns simple, low-coverage wrapper-like extraction patterns, which we then apply to conventional information extraction problems using boosting. The result is BWI, a trainable information extraction system with a strong precision bias and F1 performance better than state-of-the-art techniques in many domains.

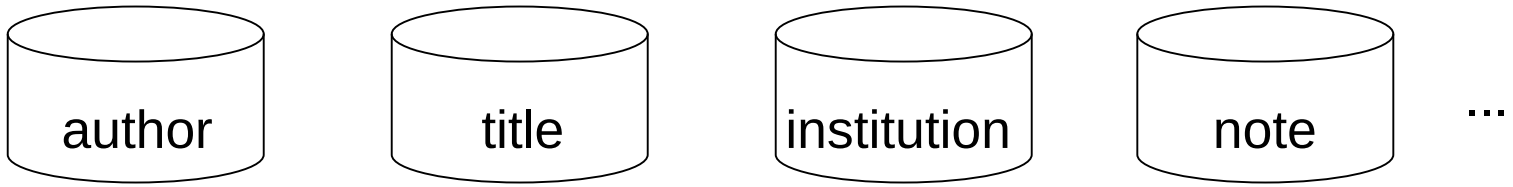
Introduction

ing email, Usenet posts, and Web pages, rely on extra-linguistic structures, such as HTML tags, document formatting, and ungrammatical stereotypic language, to convey essential information. Much recent work in IE, therefore, has focused on learning approaches that do not require linguistic information, but that can exploit other kinds of regularities. To this end, several distinct rule-learning algorithms (Soderland 1999; Califf 1998; Freitag 1998) and multi-strategy approaches (Freitag 2000) have been shown to be effective. Recently, statistical approaches using hidden Markov models have achieved high performance levels (Leek 1997; Bikel *et al.* 1997; Freitag and McCallum 1999).

At the same time, work on information integration (Wiederhold 1996; Levy *et al.* 1998) has led to a need for specialized *wrapper* procedures for extracting structured information from database-like Web pages. Recent research (Kushmerick *et al.* 1997; Kushmerick 2000; Hsu and Dung 1998; Muslea *et al.* 2000) has shown that wrappers can be automatically learned for many kinds of highly regular documents, such as Web pages generated by CGI scripts. These *wrapper induction* techniques learn simple but highly accurate contextual patterns, such as "to retrieve a

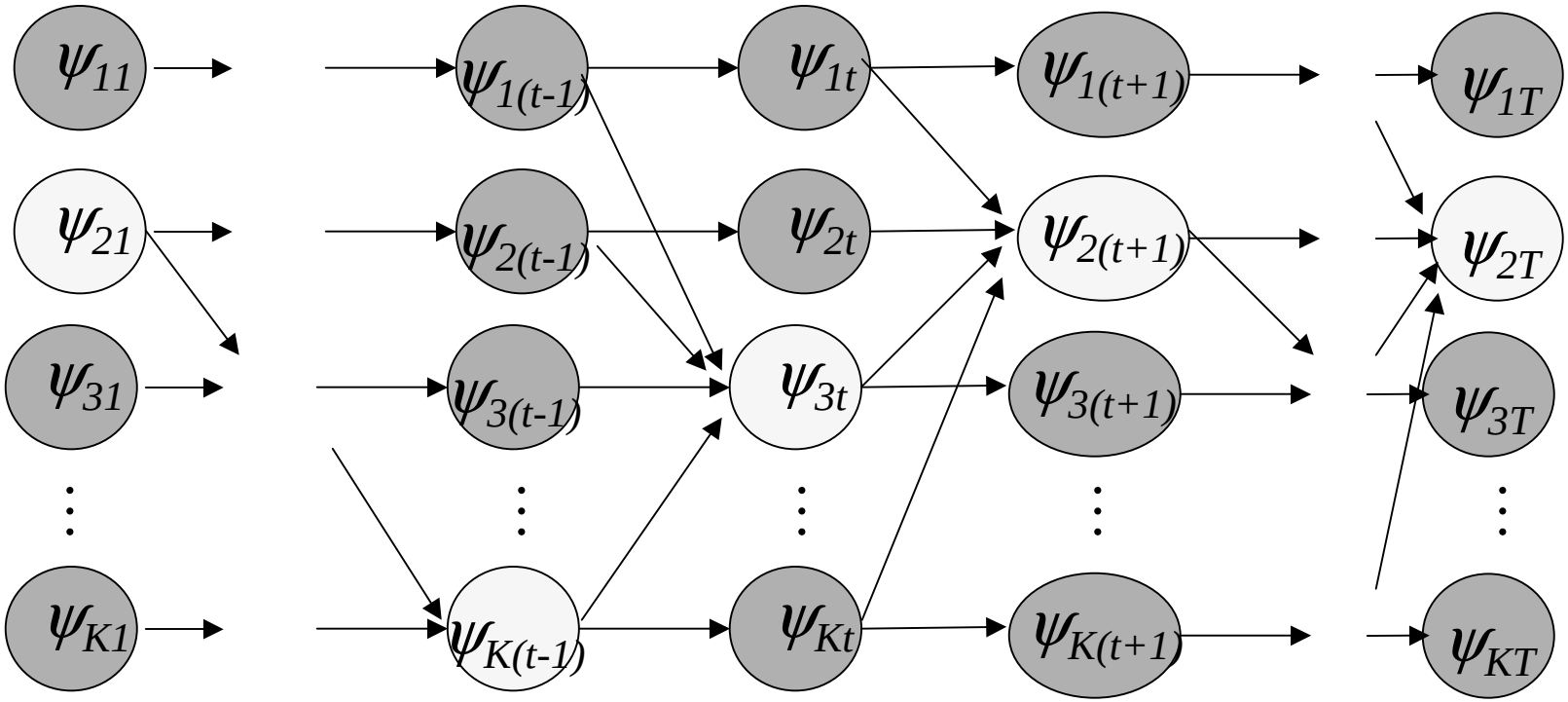
HMM for research papers: emissions [Seymore *et al.*, 99]





Trained on 2 million words of BibTeX data from the Web

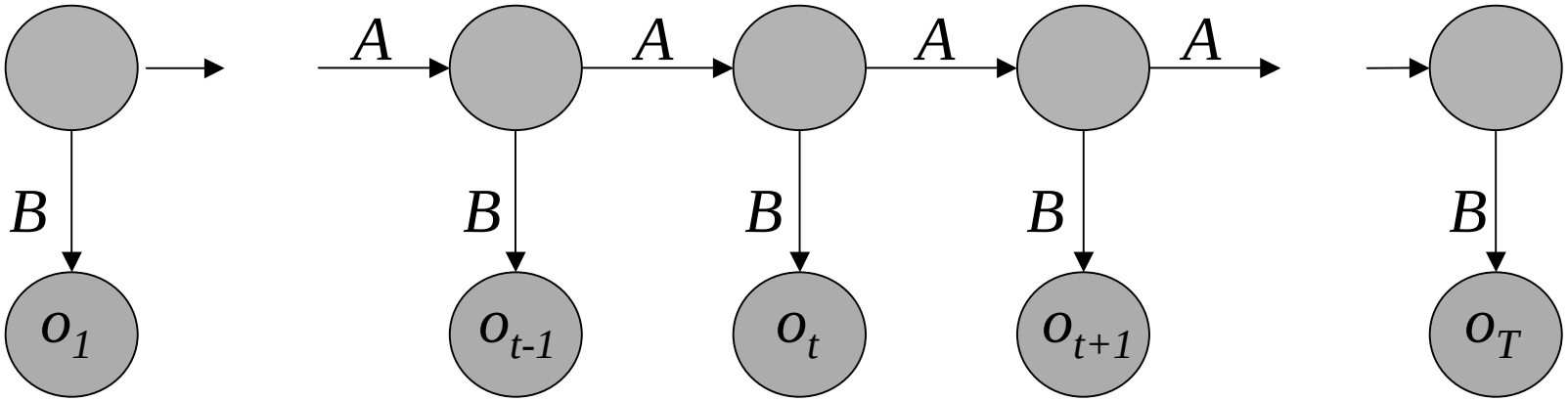
Best State Sequence: Viterbi Algorithm, Trellis View



$$\delta_j(t+1) = \max_i \delta_i(t) a_{ij} b_{jo_{t+1}}$$
$$\psi_j(t+1) = \arg\max_i \delta_i(t) a_{ij} b_{jo_{t+1}}$$

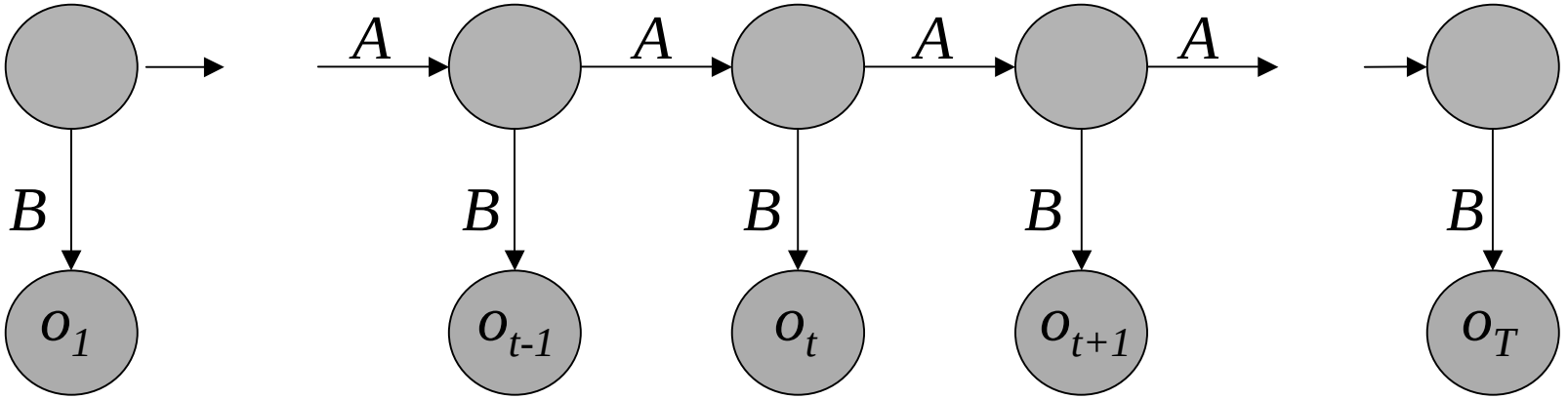
Find biggest at last time, and then trace backwards

Learning = Parameter Estimation: EM/Forward-Backward algorithm



- Given an observation sequence, find the model that is most likely to produce that sequence.
  - Find parameters so $P(O|\Theta)$ is maximized
- No analytic method, so:
- Given a model and observation sequence, update the model parameters to better fit the observations: hill climb so $P(O|\Theta)$ goes up.

Parameter Estimation: Baum-Welch or Forward-Backward



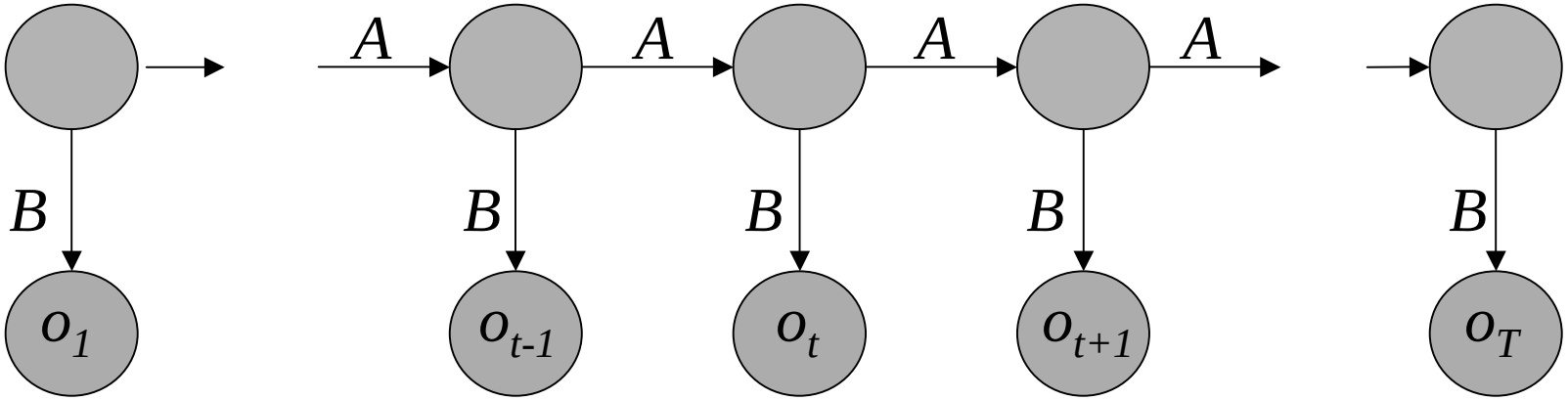
$$p_t(i,j) = \frac{\alpha_i(t) a_{ij} b_{jo_{t+1}} \beta_j(t+1)}{\sum_{m=1\ldots N} \alpha_m(t)\beta_m(t)}$$

Probability of traversing an arc

$$\gamma_i(t) = \sum_{j=1\ldots N} p_t(i,j)$$

Probability of being in state *i*

Parameter Estimation: Baum-Welch or Forward-Backward



Use ratio of expec-tations

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T} p_t(i,j)}{\sum_{t=1}^{T} \gamma_i(t)}$$

$$\hat{b}_{ik} = \frac{\sum_{\{t:o_t=k\}} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_i(t)}$$

$$\hat{\pi}_i = \gamma_i(1)$$

Now we can compute the new estimates of the model parameters.

Is it that all there is to it?

- As often with text, the biggest problem is the *sparseness* of observations (words)
- Need to use many techniques to do it well:
  - *Smoothing* (as in NB) to give suitable nonzero probability to unseens
  - *Featural decomposition* (capitalized?, number?, etc.) gives a better estimate
  - *Shrinkage* allows pooling of estimates over multiple states of same type (e.g., prefix states)
  - Well designed (or learned) HMM topology
  - *Partially annotated data:* constrained EM

HMM example

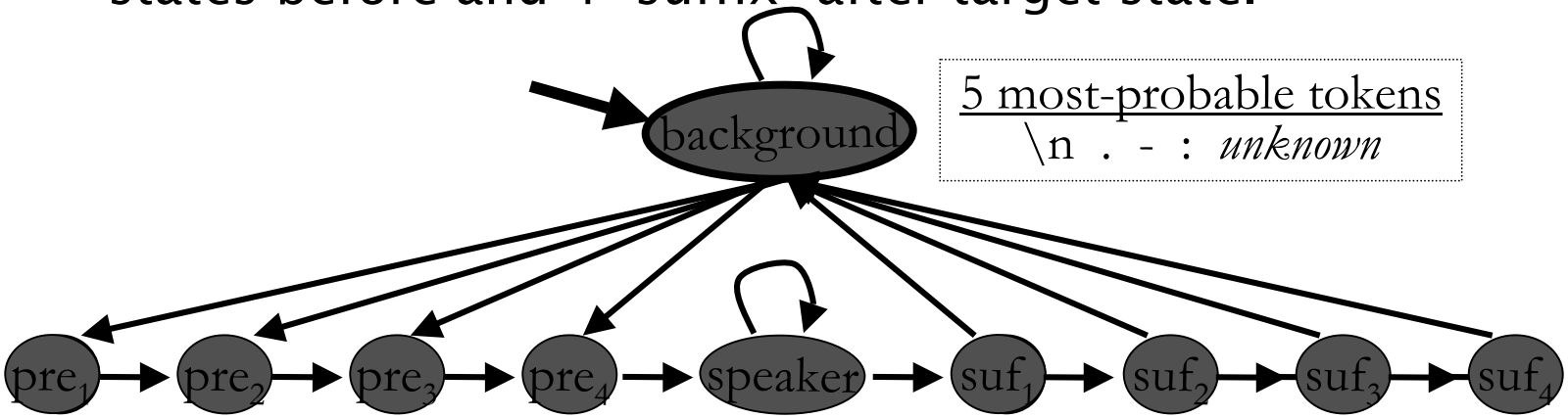
“Seminar announcements” task

<0.15.4.95.15.11.55.rudibear+@CMU.EDU.0>
Type: cmu.andrew.assocs.UEA
Topic: Re: entreprenuership speaker
Dates: 17-Apr-95
Time: 7:00 PM
PostedBy: Colin S Osburn on 15-Apr-95 at 15:11 from CMU.EDU
Abstract:

hello again
to reiterate
there will be a speaker on the law and startup business
this **monday evening the 17th**
it will be at **7pm** in **room 261 of GSIA** in the new building, ie upstairs.
please attend if you have any interest in starting your own business or
are even curious.
Colin

HMM example, continued

Fixed topology that captures limited context: 4 "prefix" states before and 4 "suffix" after target state.



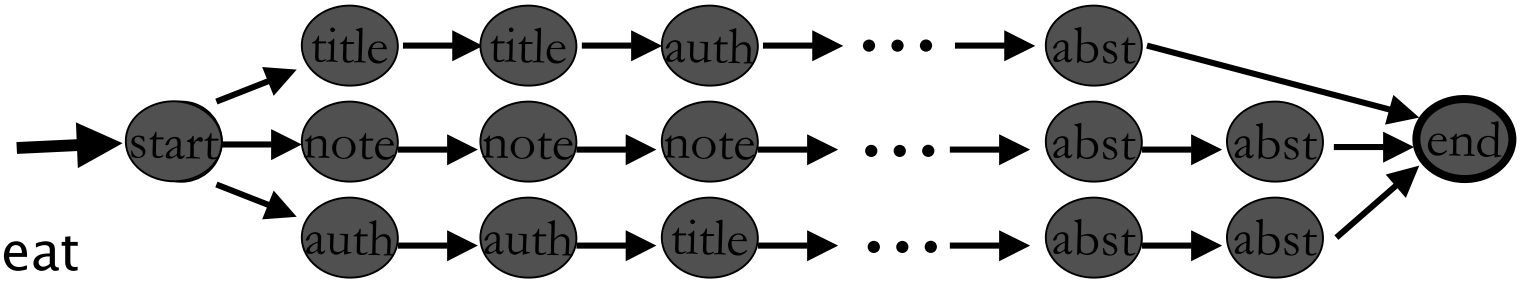
5 most-probable tokens
\n . - : *unknown*

\n	\n	\n	\n	*unknown*	\n	\n	\n	\n
seminar	:	who	:	.	,	*unknown*	of	of
.	.	speaker	.	dr	will	.	.	*unknown*
robotics	-	:	with	professor	(	department	,	.
unknown	*unknown*	.	,	michael	-	the	*unknown*	:

[Freitag, 99]

Learning HMM structure [Seymore *et al*, 1999]

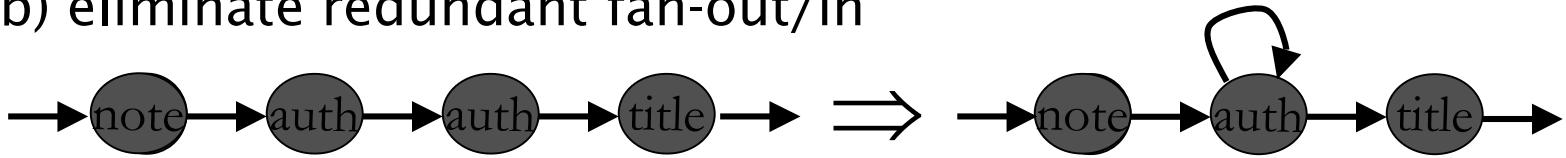
start with maximally-specific HMM (one state per observed word):



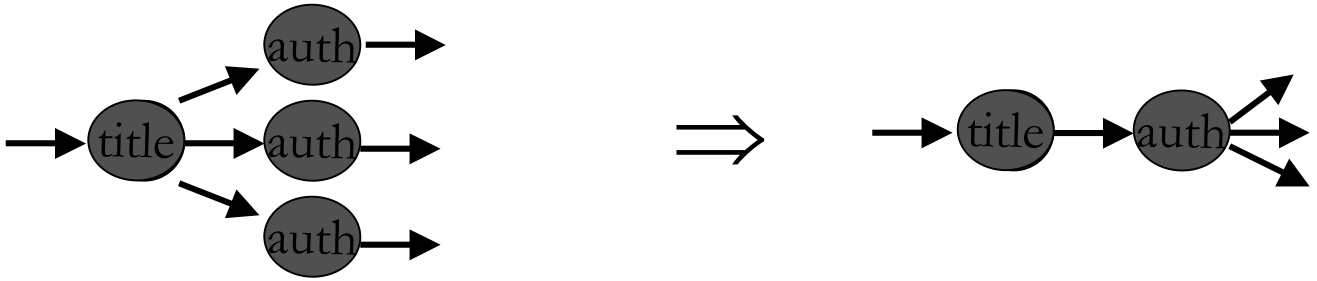
repeat

(a) merge adjacent identical states

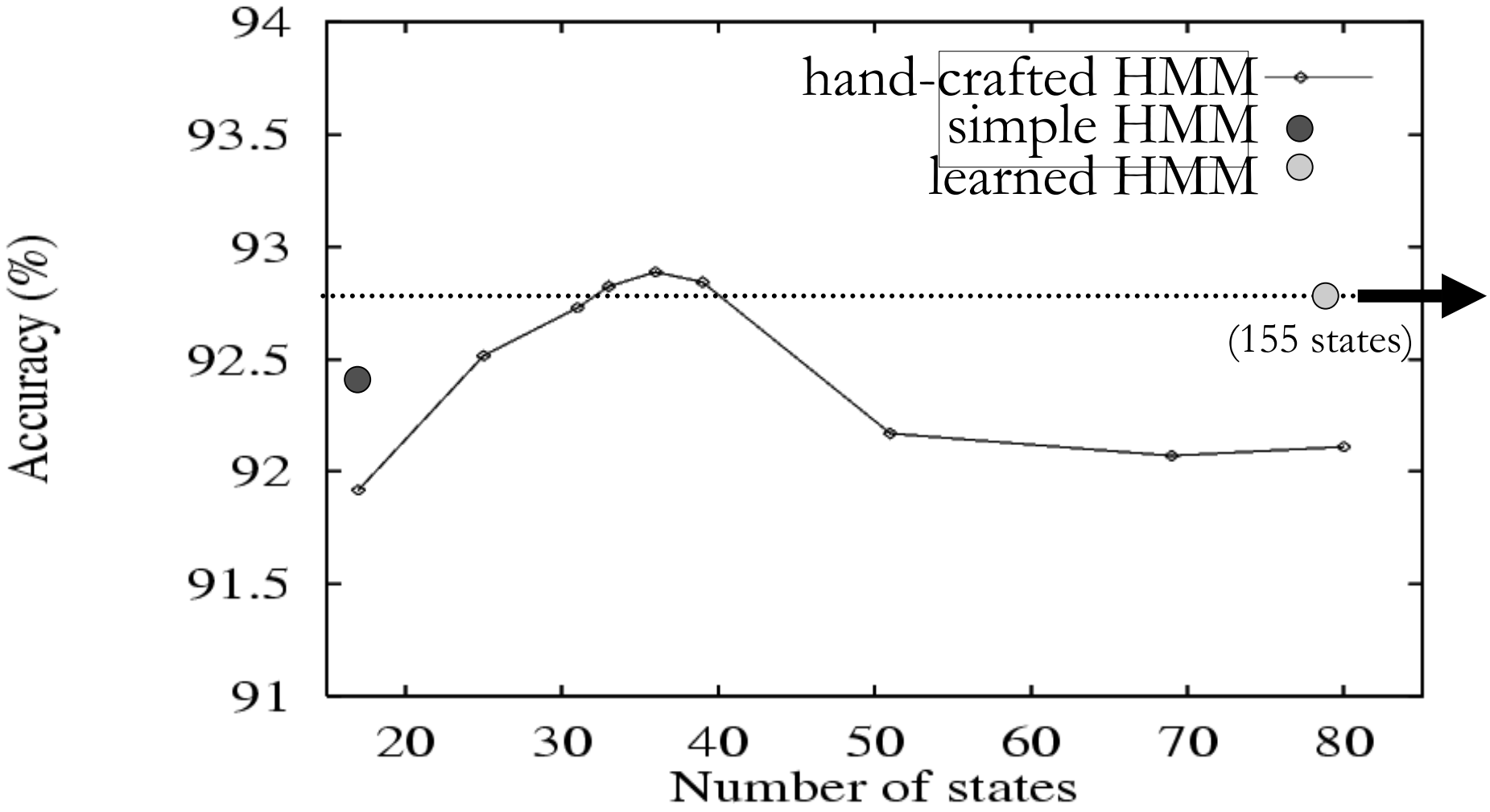
(b) eliminate redundant fan-out/in



until obtain good tradeoff between HMM accuracy and complexity



Evaluation (% tokens tagged correctly)



References

- Mary Elaine Califf and Raymond J. Mooney: Relational Learning of Pattern-Match Rules for Information Extraction. In AAAI 1999: 328-334.
- Leek, T. R. 1997, Information Extraction using Hidden Markov Models, Master's thesis, UCSD
- Bikel, D. M.; Miller, S; Schwartz, R.; and Weischedel, R. 1997, Nymble: a high-performance learning name-finder. In *Proceedings of ANLP-97*, 194-201. [Also in MLJ 1999]
- Kristie Seymore, Andrew McCallum, Ronald Rosenfeld, 1999, Learning Hidden Markov Model Structure for Information Extraction, In *Proceedings if the AAAI-99 Workshop on ML for IE*.
- Dayne Freitag and Andrew McCallum, 2000, Information Extraction with HMM Structures Learned by Stochastic Optimization. *AAAI-2000*.

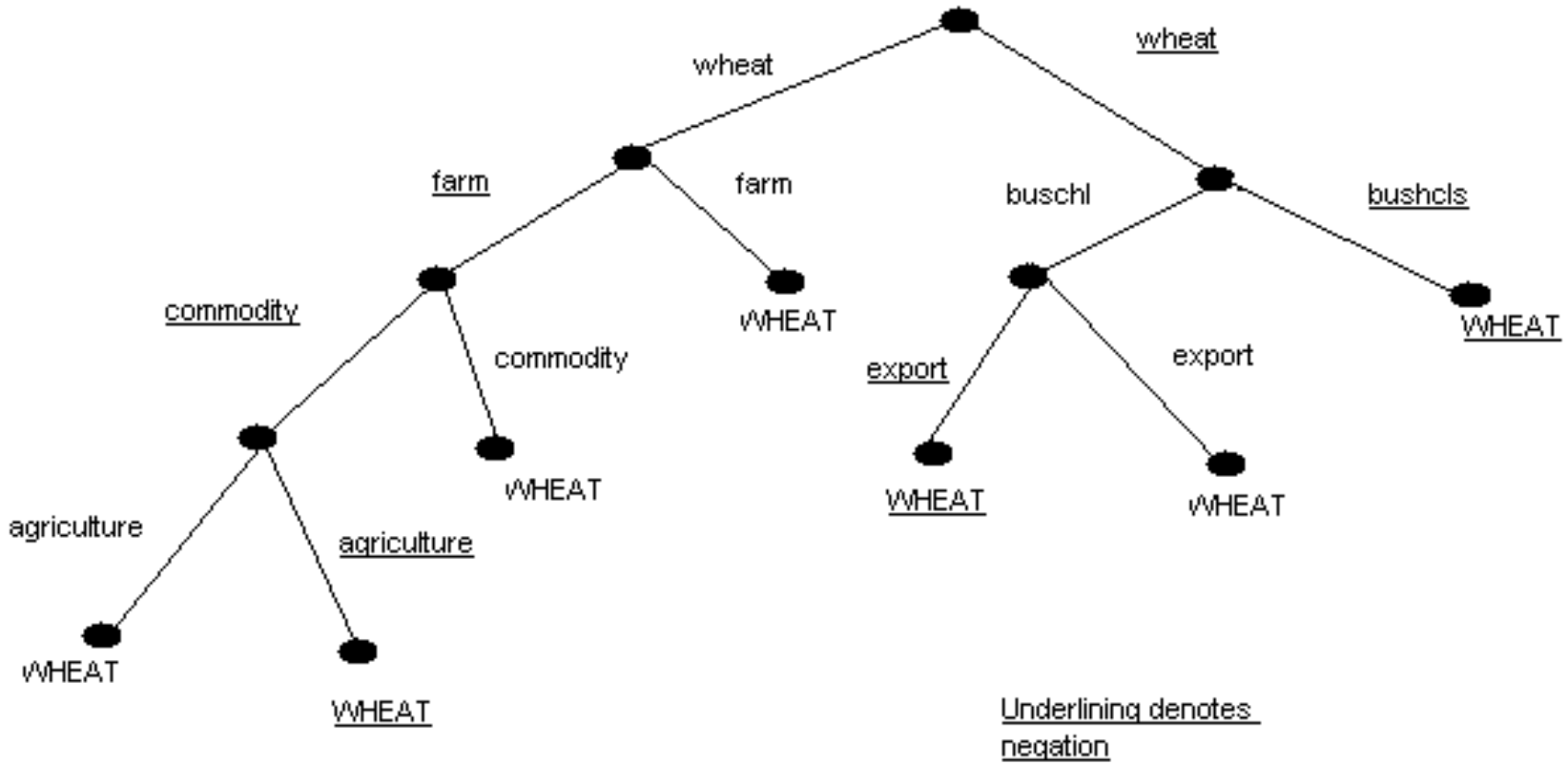
Text Categorization

- *Today:*
  - *Decision Trees*
  - *Logistic Regression models*
  - *Metadata via classification/IE*
- *Thursday:*
  - *Nearest neighbor methods*
  - *Support vector machines*
  - *Hypertext categorization*

Decision Trees

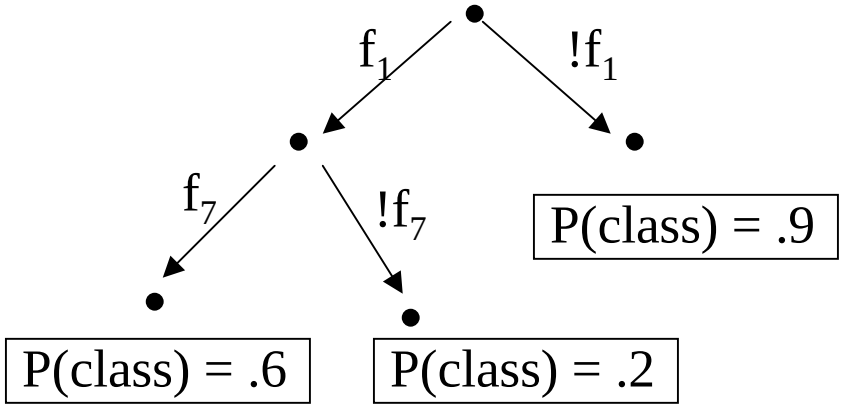
- Tree with internal nodes labeled by terms
- Branches are labeled by tests on the weight that the term has
- Leaves are labeled by categories
- Classifier categorizes document by descending tree following tests to leaf
- The label of the leaf node is then assigned to the document
- Most decision trees are binary trees (never disadvantageous; may require extra internal nodes)

Decision Tree Example



Decision Tree Learning

- Learn a sequence of tests on features, typically using top-down, greedy search
  - At each stage choose unused feature with highest Information Gain (as in Lecture 5)
- Binary (yes/no) or continuous decisions



Decision Tree Learning

- Fully grown trees tend to have decision rules that are overly specific and are therefore unable to categorize documents well
  - Therefore, pruning or early stopping methods for Decision Trees are normally a standard part of classification packages
- Use of small number of feature test potentially bad in text cat, but in practice method does well
- Decision trees are very easily interpreted by humans – much more easily than probabilistic methods like Naive Bayes
- Decision Trees are normally regarded as a symbolic machine learning algorithm, though can be used probabilistically

Recall: Classic Reuters Data Set (21578 - ModApte split)

- 9603 training, 3299 test articles; ave. 200 words
- 118 categories
  - An article can be in more than one category
  - Learn 118 binary category distinctions
- 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

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)

Dumais et al. 1998: Reuters - Accuracy ((R+P)/2)

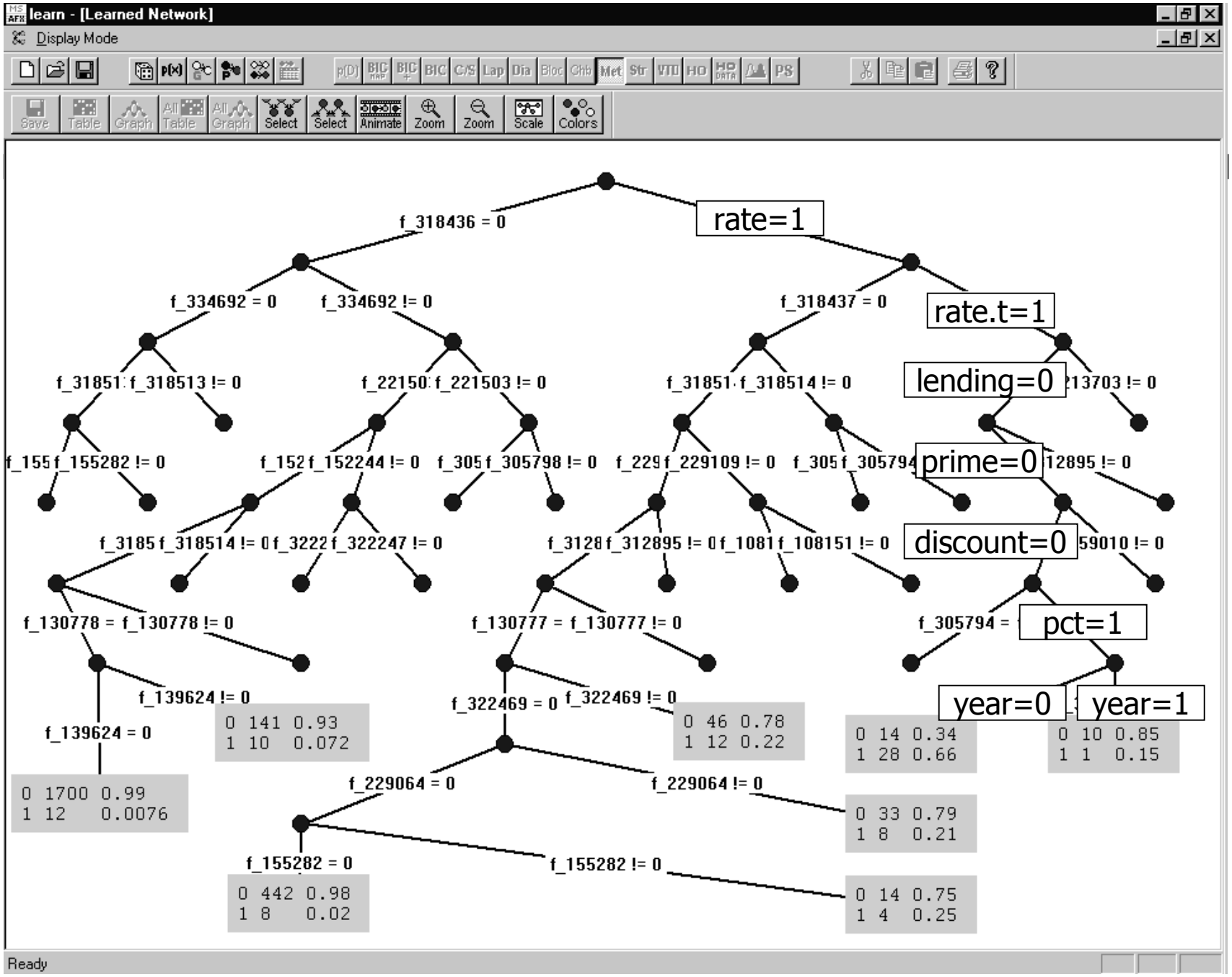
	Findsim	NBayes	BayesNets	Trees	LinearSVM
earn	92.9%	95.9%	95.8%	97.8%	98.2%
acq	64.7%	87.8%	88.3%	89.7%	92.8%
money-fx	46.7%	56.6%	58.8%	66.2%	74.0%
grain	67.5%	78.8%	81.4%	85.0%	92.4%
crude	70.1%	79.5%	79.6%	85.0%	88.3%
trade	65.1%	63.9%	69.0%	72.5%	73.5%
interest	63.4%	64.9%	71.3%	67.1%	76.3%
ship	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

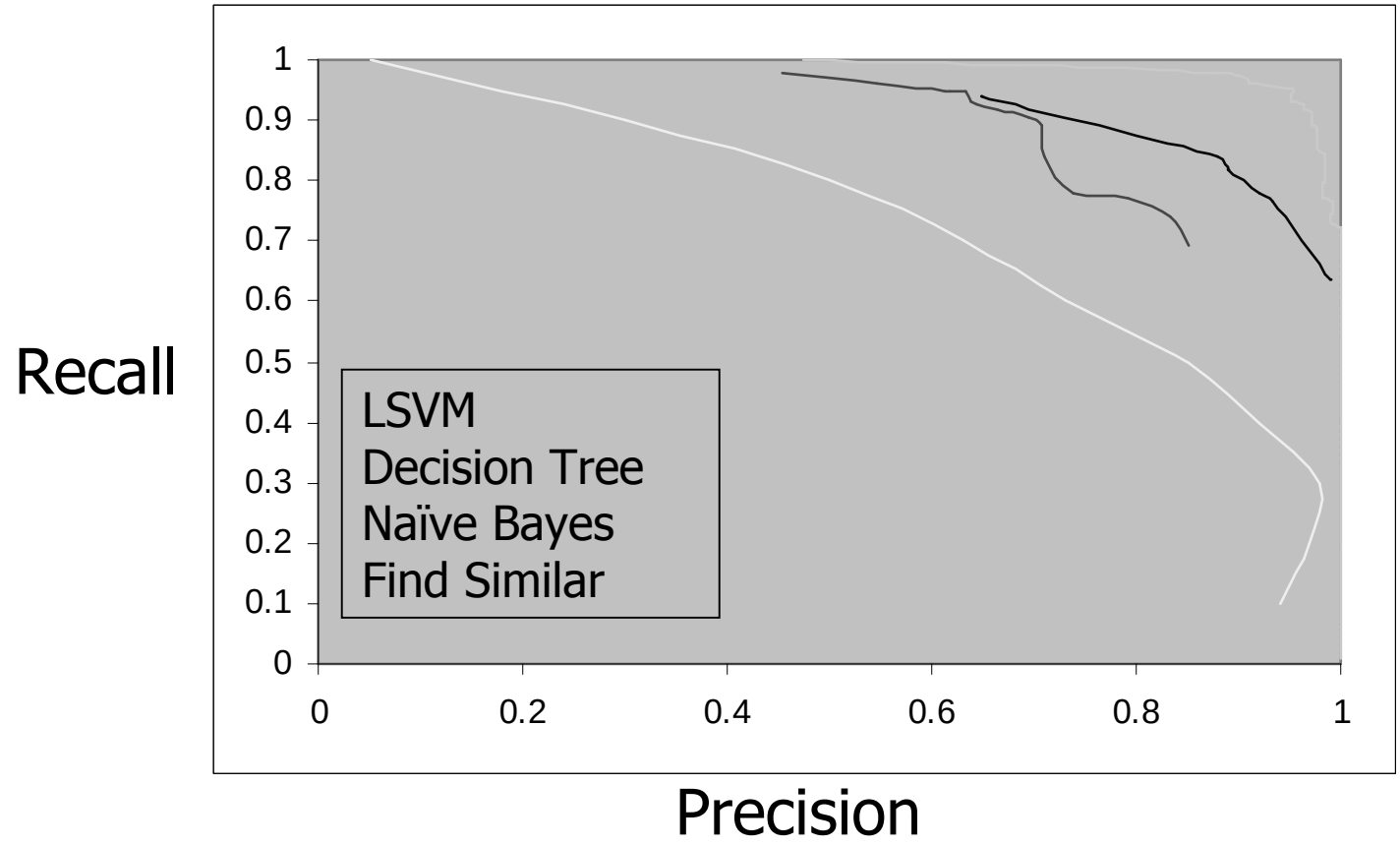
Precision: % really in category among those stories labeled in category

Break Even: (Recall + Precision) / 2

Category: “interest”



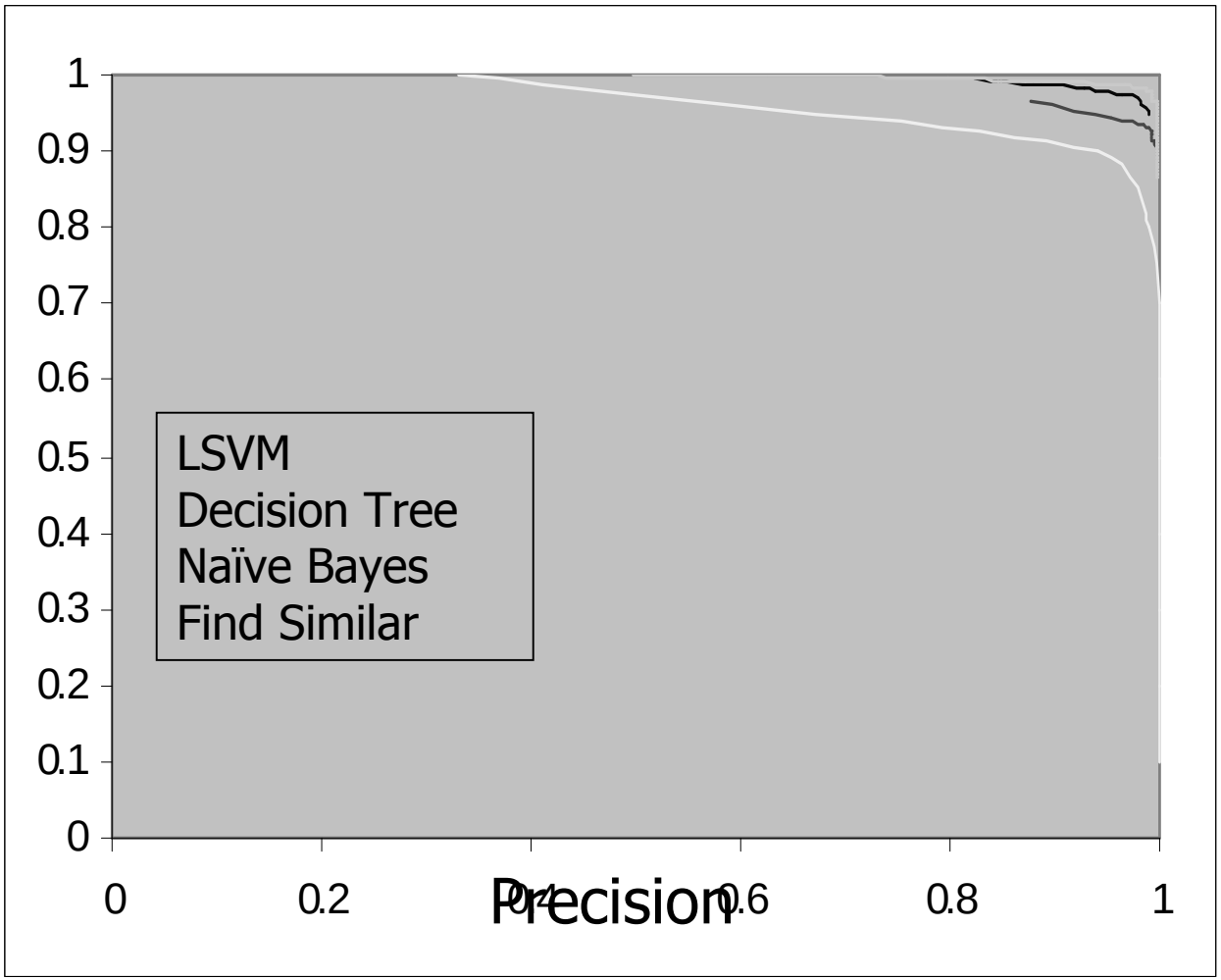
Reuters ROC - Category Grain



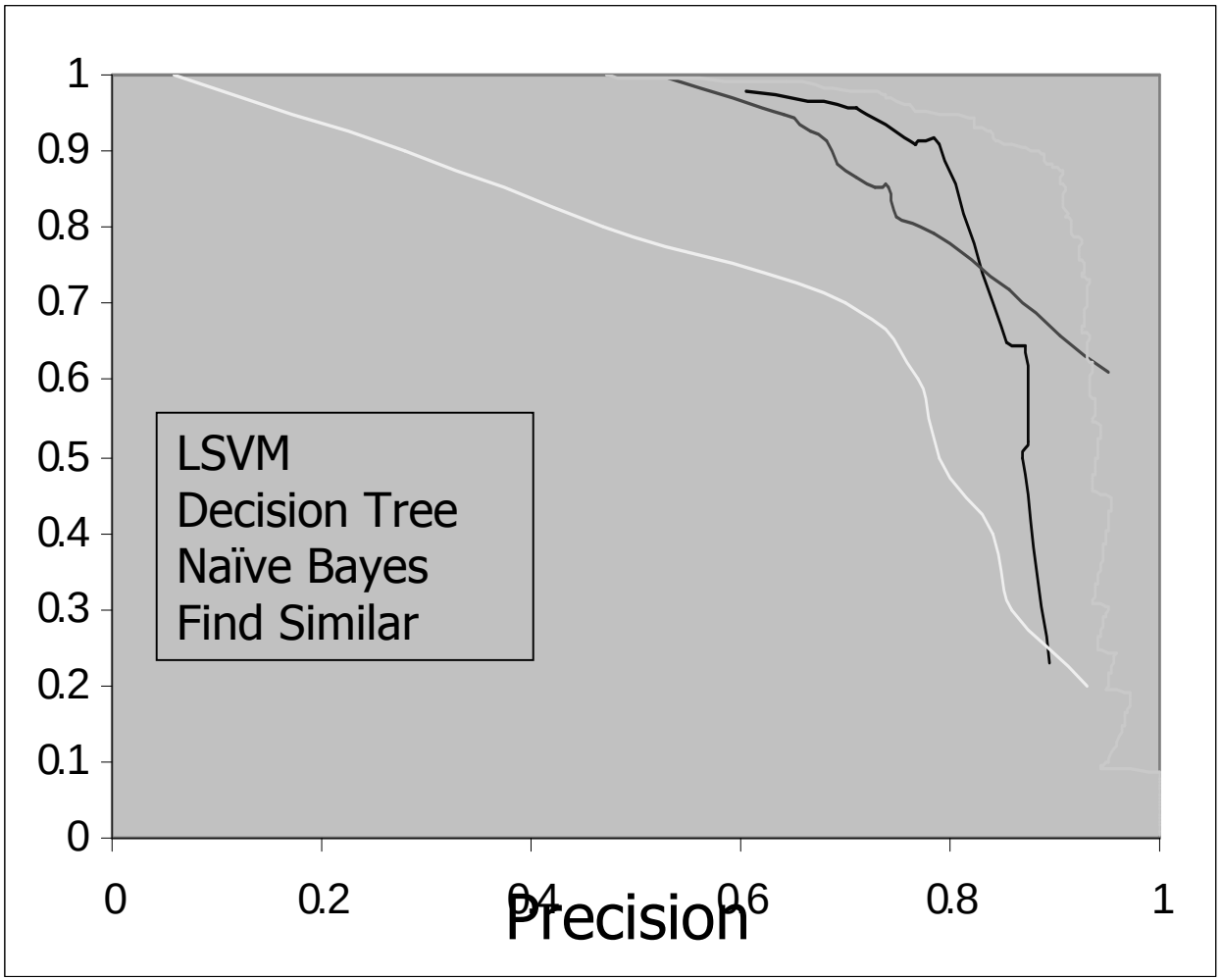
Recall: % labeled in category among those stories that are really in category

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

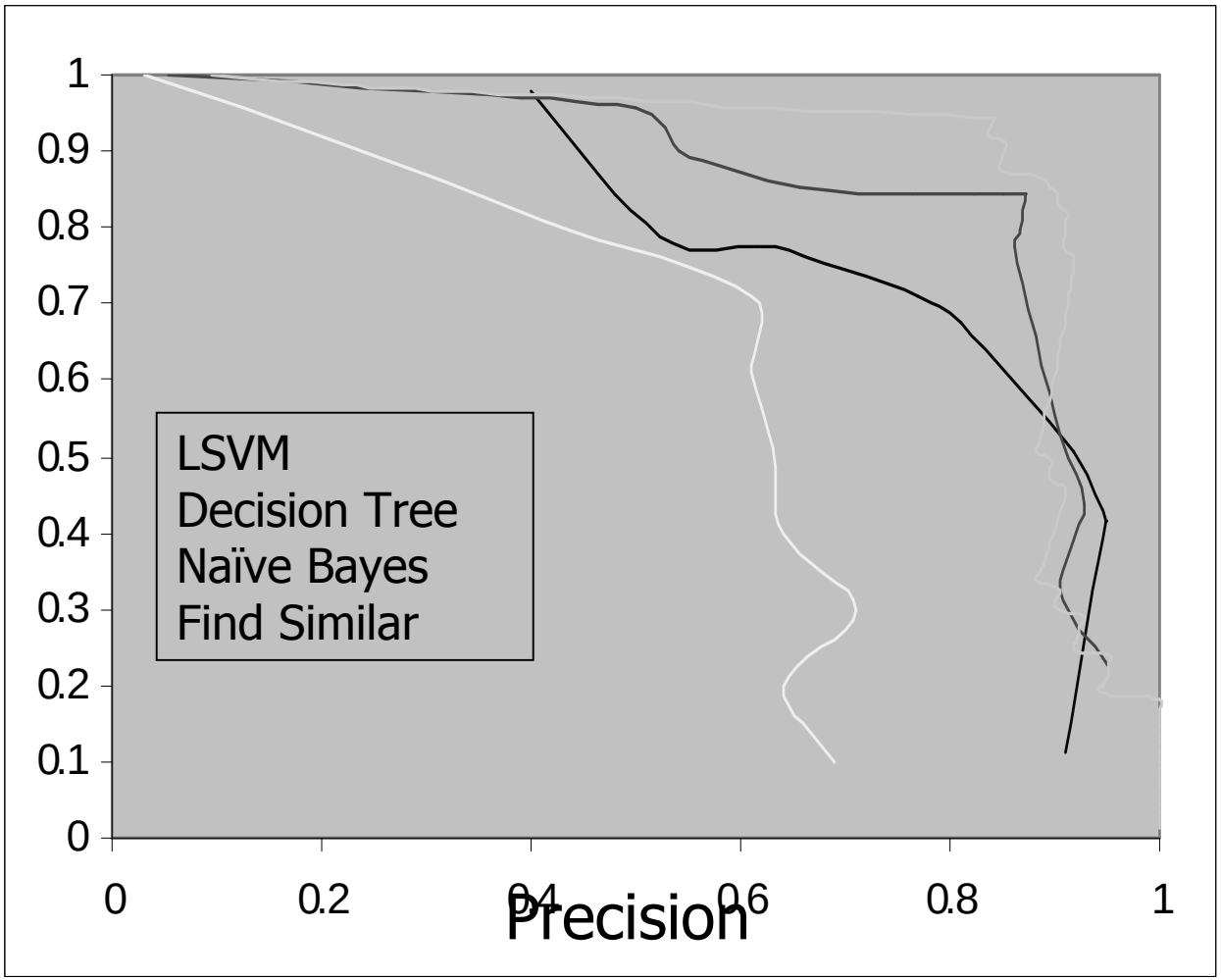
ROC for Category - Earn



ROC for Category - Crude



ROC for Category - Ship



Intro to logistic regression

- Naïve Bayes and probabilistic language models for IR are *generative models*
- Model predicts probability document *d* will be generated by a source *c*
- e.g. Naïve Bayes unigram language model:

$$P(d \mid c) = \prod_{w \in d} P(w \mid c)$$

- Parameters, i.e. *P(w|c)*'s, are fit to optimally predict generation of *d*

Classify Text w/ Gen. Model

- One source model for each class *c*
- Choose class *c* with largest value of:

$$P(c \mid d) = P(d \mid c) \cdot P(c) / k$$

$$\log P(c \mid d) = \log P(c) + \sum_{w \in d} \log P(w \mid c)$$

- For 2 classes, unigram *P(d|c)*, we have:

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

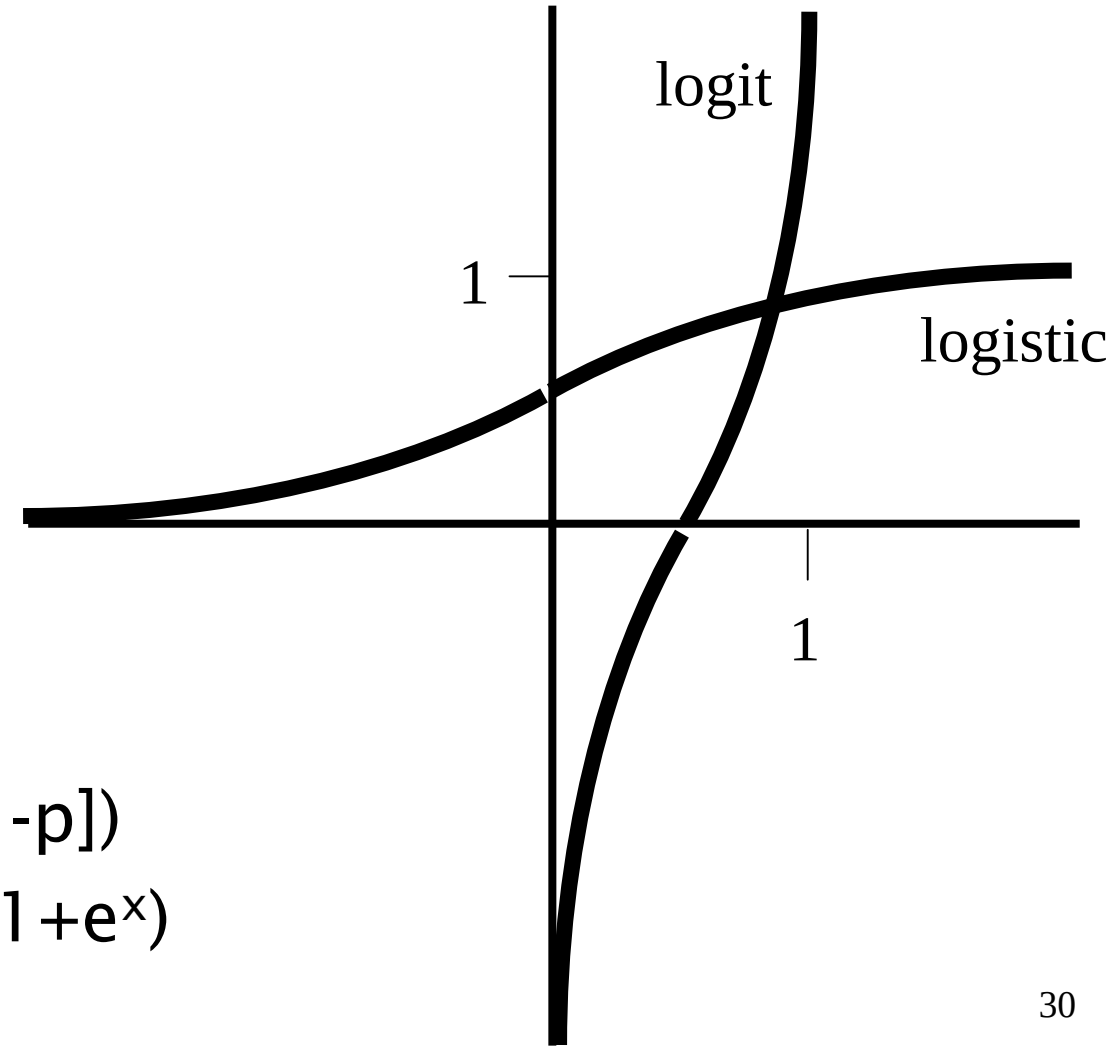
The discriminative alternative: Logistic Regression

- Directly model probability of generating *class* conditional on words: *P(c|w)*
- Logistic regression:

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

- Tune parameters *w* to optimize *conditional* likelihood (class probability predictions)
- What a statistician would probably tell you to use if you said you had a categorical decision problem (like text categorization)

Logit and logistic transforms



logit(p) = ln(p/[1-p])

logistic(x) = $e^x/(1+e^x)$

The Logit-based Model

- logit(answer) = logit(F1) + logit(F2) + logit(F3) + ... + logit(Fn)
- Model: The logit of a predicted probability is sum of the logits of the probabilities associated with each of the features active for the data point
- An instance of a *generalized linear model* where one response conditioned on all features
- A.k.a. maximum entropy models, exponential models, conditional markov random fields, Gibbs distributions, ….

Interpreting Coefficients

- Since:

$$\ln[p/(1-p)] = \alpha + \beta X + e$$

- The slope coefficient (β) is interpreted as the rate of change in the log odds as X changes
- A more intuitive interpretation of the logit coefficient is the "odds ratio":
- Since

$$[p/(1-p)] = \exp(\alpha + \beta X)$$

- $\exp(\beta)$ is the effect of the independent variable on the odds of having a certain classification

(Conditional) Maximum Likelihood Estimation (MLE)

- MLE is a statistical method for estimating the coefficients of a model that maximizes some likelihood
- Here the likelihood function (L) measures the probability of observing the particular set of dependent variable values (c_1, c_2, ..., c_n) that occur in the training data
- LogR MLE is normally done by some form of iterative fitting algorithm, or a gradient descent procedure such as CG
  - Expensive for large models with many features

LR & NB: Same Parameters!

$$\text{LR}: \log \frac{P(C \mid d)}{P(\overline{C} \mid d)} = \alpha + \sum_{w \in d} \beta_w \times w$$

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

for 2- or k-class, binary or raw TF weighting …
but optimized differently

Performance

- Early results with LogR were disappointing, because people didn't understand the means to *regularize* (smooth) LogR to cope with sparse data
- Done right, LogR outperforms NB in text categorization and batch filtering studies
  - NB optimizes parameters to predict words, LR optimizes to predict *class*
- LogR seems as good as SVMs (or any known text cat method – Tong & Oles 2001) though less studied and less trendy than SVMS.

Metadata:
Opportunities for TextCat/IE

Why metadata?

- Metadata = “data about data”
- “Normalized” semantics
- Enables easy searches otherwise not possible:
  - Time
  - Author
  - Url / filename
- And gives information on non-text content
  - Images
  - Audio
  - Video

For Effective Metadata We Need:

- Semantics
  - Commonly understood terms to describe information resources
- Syntax
  - Standard grammar for connecting terms into meaningful “sentences”
- Exchange framework
  - So we can recombine and exchange metadata across applications and subjects

Dublin Core Element Set

- Title (e.g., Dublin Core Element Set)
- Creator (e.g., Hinrich Schuetze)
- Subject (e.g, keywords)
- Description (e.g., an abstract)
- Publisher (e.g., Stanford University)
- Contributor (e.g., Chris Manning)
- Date (e.g, 2002.12.03)
- Type (e.g., presentation)
- Format (e.g., ppt)
- Identifier (e.g., http://www.stanford.edu/class/cs276a/syllabus.html)
- Source (e.g. http://dublincore.org/documents/dces/)
- Language (e.g, English)
- Coverage (e.g., San Francisco Bay Area)
- Rights (e.g., Copyright Stanford University)

Dublin Core: Goals

- Metadata standard
- Framework for interoperability
- Facilitate development of subject specific metadata
- More general goals of DC community
  - Foster development of metadata tools
    - Creating, editing, managing, navigating metadata
  - Semantic registry for metadata
    - Search declared meanings and their relationships
  - Registry for metadata vocabularies
    - Maybe somebody has already done the work

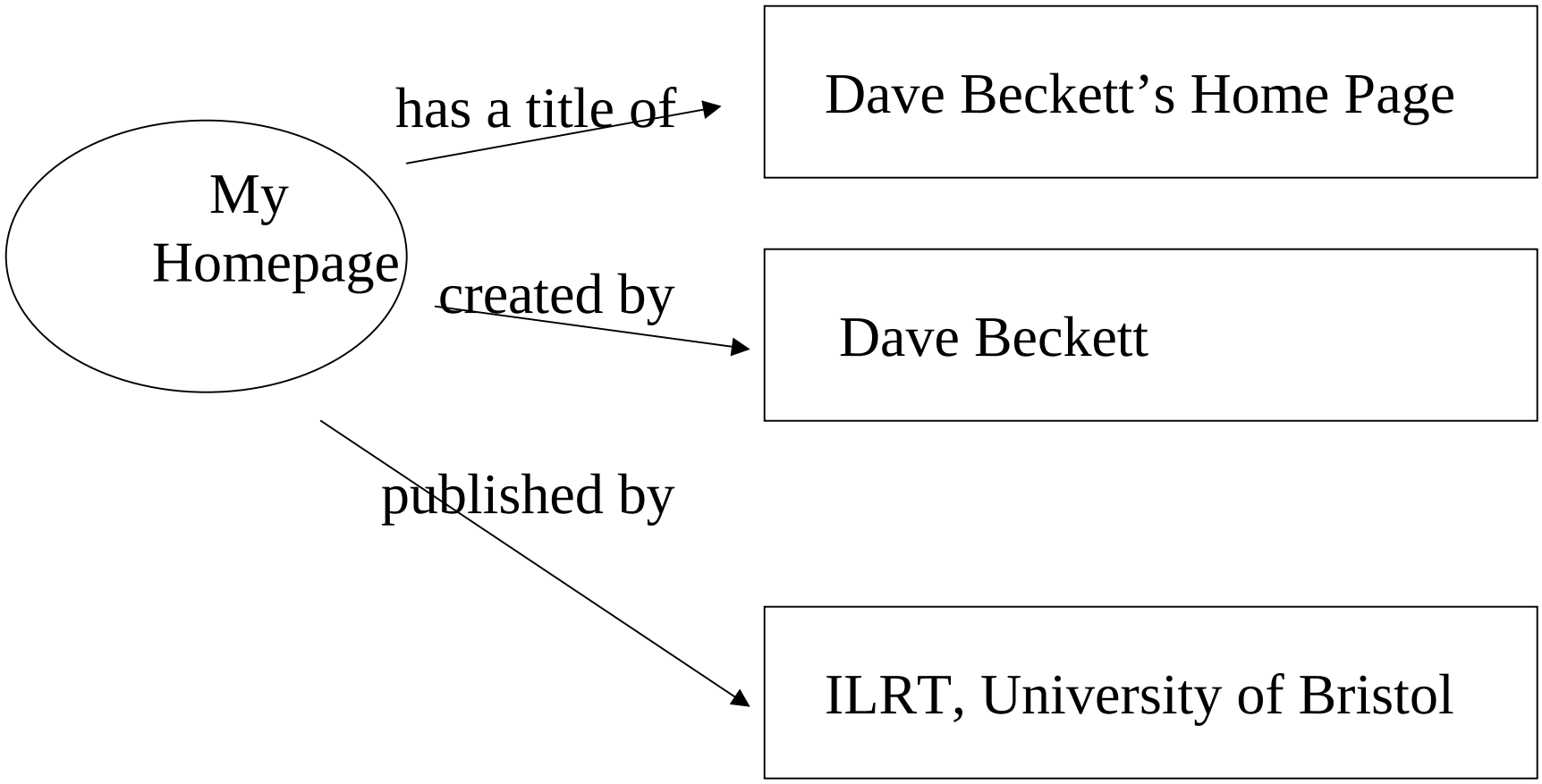
RDF = Resource Description Framework

- Emerging standard for metadata
- W3C standard
  - Part of W3C's metadata framework
- Specialized for WWW
- Desiderata
  - Combine different metadata modules (e.g., different subject areas)
  - Syndication, aggregation, threading

RDF example in XML

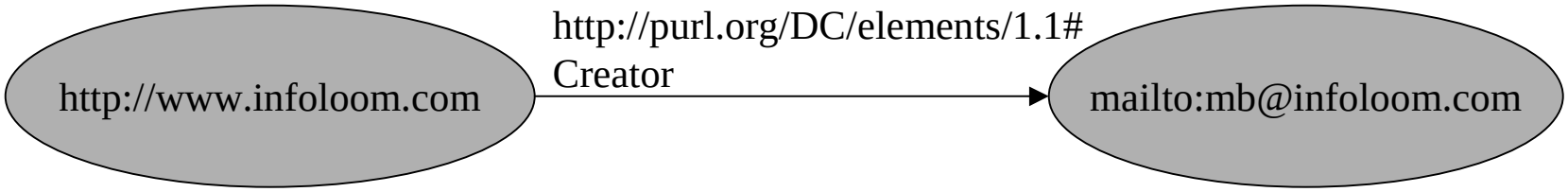
```
<?xml version="1.0"?> <rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#" xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description
rdf:about="http://www.ilrt.org/people/cmdjb/">
<dc:title>Dave Beckett's Home Page</dc:title>
<dc:creator>Dave Beckett</dc:creator>
<dc:publisher>ILRT, University of
Bristol</dc:publisher> </rdf:Description> </rdf:RDF>
```

RDF example


has a title of
Dave Beckett's Home Page
My Homepage
created by
Dave Beckett
published by
ILRT, University of Bristol

Resource Description Framework (RDF)

- RDF was conceived as a way to wrap metadata assertions (eg Dublin Core information) around a web resource.
- The central concept of the RDF data model is the *triple*, represented as a labeled edge between two nodes.
- The subject, the object, and the predicate are all resources, represented by URIs



- Properties can be multivalued for a resource, and values can be literals instead of resources
- Graph pieces can be chained and *nested*
- RDF Schema gives frame-based language for ontologies and reasoning over RDF.

Metadata Pros and Cons

- CONS
  - Most authors are unwilling to spend time and energy on
    - learning a metadata standard
    - annotating documents they author
  - Authors are unable to foresee all reasons why a document may be interesting.
  - Authors may be motivated to sabotage metadata (patents).
- PROS
  - Information retrieval often does not work.
  - Words poorly approximate meaning.
  - For truly valuable content, it pays to add metadata.
- Synthesis
  - In reality, most documents have some valuable metadata
  - If metadata is available, it improves relevance and user experience
  - But most interesting content will always have inconsistent and spotty metadata coverage

Metadata and TextCat/IE

- The claim of metadata proponents is that metadata has to be explicitly annotated, because we can't hope to get, say, a book price from varied documents like:

<H1>
<The Rhyme of the Ancient Mariner>
</H1>
The Rhyme of the Ancient Mariner, by Samuel Coleridge, is available for the low price of $9.99. This Dover reprint is beautifully illustrated by Gustave Dore.
<p>
Julian Schnabel recently directed a movie, Pandemonium, about the relationship between Coleridge and Wordsworth.

Metadata and TextCat/IE

- … but with IE/TextCat, these are exactly the kind of things we *can* do
- Of course, we can do it more accurately with human authored metadata
  - But, of course, the metadata might not match the text (*metadata spamming*)
- Opens up an interesting world where agents use metadata if it's there, but can synthesize it if it isn't (by text cat/IE), and can verify metadata for correctness against text
  - Seems a promising area; not much explored!

References

- 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.
- Christopher Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing.* Cambridge, MA: MIT Press
- Zhang Tong and Frank J. Oles. 2001. Text Categorization Based on Regularized Linear Classification Methods. *Information Retrieval* 4(1): 5-31.
- 'Classic' Reuters data set: http://www.daviddlewis.com /resources /testcollections/reuters21578/
- Tim Berners Lee on semantic web: http://www.sciam.com/ 2001/0501issue/0501berners-lee.html
- http://www.xml.com/pub/a/2001/01/24/rdf.html