Discriminative Models in NLP

CS224N/Ling237

April 30, 2003

Overview

- Conditional Log-linear (Maximum Entropy) Models
  - Toy example
  - Parameter estimation
- Maximum Entropy Markov Models (Discriminative Sequence Models) for IE
- Some Theory of Generative/Discriminative Models for Classification
- Empirical Comparison (Word Sense Disambiguation and Part-of-Speech Tagging)

Conditional Log-linear Models

- A class of models increasingly popular and successful in machine learning of natural language
- Not generative models like n-gram, HMM and Naive Bayes
- Diverse features can be defined, and information from them is combined in a way to best discriminate between target classes
- These models have been applied to single classification (PP attachment, Word Sense Disambiguation, etc.) and in sequence tasks (POS tagging, IE, Parsing); we will see examples of some of these

Conditional Log-linear Models

- Given a set of training data {(x_1,c_1), ...,(x_n,c_n)}, define features $f_j : X \times C \rightarrow R$
- Learn a model of the form

$$P(c \mid x) = \frac{\exp(\sum_{j=1}^{m} f_j(x,c)\lambda_j)}{\sum_{k=1}^{K} \exp(\sum_{j=1}^{m} f_j(x,c_k)\lambda_j)}$$

- Now the question is how to choose the features (f) and the parameters (λ)

Conditional Log-linear Models: Example

- We are searching the Web for documents having to do with Bush's attitudes toward energy conservation. (x=documents, c=1/0)
- Three features useful for such a task might be:

  f1(doc,c)=1, iff “President” appears in doc and c=1 (0 otherwise)

  f2(doc,c)=1, iff “conservation” appears in doc and c=1 (0 otherwise)

  f3(doc,c)=1, iff “President” and “conservation” appear in doc and c=1 (0 otherwise)
- We can define as many features as we like, depending on the words in the document; features that are conjunctions or disjunctions of other features are often useful

Fitting the Model

- To find the parameters $\lambda_1, \lambda_2, \lambda_3$

  write out the conditional log-likelihood of the training data and maximize it

$$CLogLik(D) = \sum_{i=1}^{n} \log P(c_i \mid x_i)$$

- The log-likelihood is concave and has a single maximum; use your favorite numerical optimization package

Fitting the Model Generalized Iterative Scaling

- A simple optimization algorithm which works when the features are non-negative
- We need to define a slack feature to make the features sum to a constant over all considered pairs from $X \times C$
- Define

$$M = \max_{i,c} \sum_{j=1}^{m} f_j(x_i, c)$$

- Add new feature

$$f_{m+1}(x, c) = M - \sum_{j=1}^{m} f_j(x, c)$$

Generalized Iterative Scaling

- Compute empirical expectation for all features

$$E_{\tilde{p}}(f_j) = \frac{1}{N}\sum_{i=1}^{n} f_j(x_i, c_i)$$

- Initialize $\lambda_j = 0,\ j = 1...m+1$
- Repeat
  - Compute feature expectations according to current model

$$E_{p^t}(f_j) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} P(c_k \mid x_i) f_j(x_i, c_k)$$

  - Update parameters

$$\lambda_j^{(t+1)} = \lambda_j^{(t)} + \frac{1}{M}\log\left(\frac{E_{\tilde{p}}(f_j)}{E_{p^t}(f_j)}\right)$$

- Until converged

Conditional Log-linear and Maximum Entropy Models

- These models are also called maximum entropy models
- The model having maximum entropy and satisfying the constraints

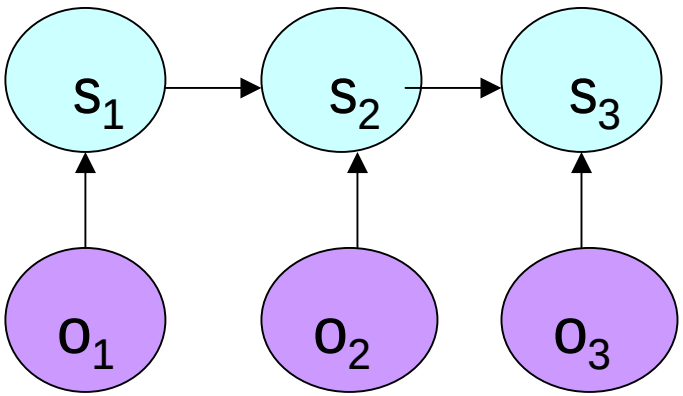
$$E_p(f_j) = E_{\tilde{p}}(f_j), \forall j$$

- Is the same model as the log-linear model of the form we saw before that maximizes the training data conditional likelihood

Discriminative Sequence Models: Maximum Entropy Markov Models for IE

A. McCallum, D. Freitag,F. Pereira (2000)

MEMMs have the the following graphical model



- The distribution of next state given previous state and current observation is estimated using a maximum entropy model
- Some slides by Fernando Pereira will be used

Problems with HMMs

- Applications need richer input representations: multiple *overlapping* features, whole chunks of text

Word features	*Line features*
word identity	centered
capitalization	indentation
ends in "-tion"	white space ratio
word in word list	begins with number
word font	ends with "?"

- Generative models do not handle easily overlapping, non-independent features
- Alternative: *conditional* model $P(\mathbf{s}|\mathbf{o})$

Application: Q-A pairs from FAQ

X-NNTP-Poster: NewsHound v1.33
Archive-name: acorn/faq/part2
Frequency: monthly

2.6) What configuration of serial cable should I use?

Here follows a diagram of the necessary connections for common terminal programs to work properly. They are as far as I know the informal standard agreed upon by commercial comms software developers for the Arc.

Pins 1, 4, and 8 must be connected together inside the 9 pin plug. This is to avoid the well known serial port chip bugs. The modem's DCD (Data Carrier Detect) signal has been re-routed to the Arc's RI (Ring Indicator) most modems broadcast a software RING signal anyway, and even then it's really necessary to detect it for the model to answer the call.

2.7) The sound from the speaker port seems quite muffled.
How can I get unfiltered sound from an Acorn machine?

All Acorn machine are equipped with a sound filter designed to remove high frequency harmonics from the sound output. To bypass the filter, hook into the Unfiltered port. You need to have a capacitor. Look for LM324 (chip 39) and and hook the capacitor like this:

Application Q-A Pairs from FAQ

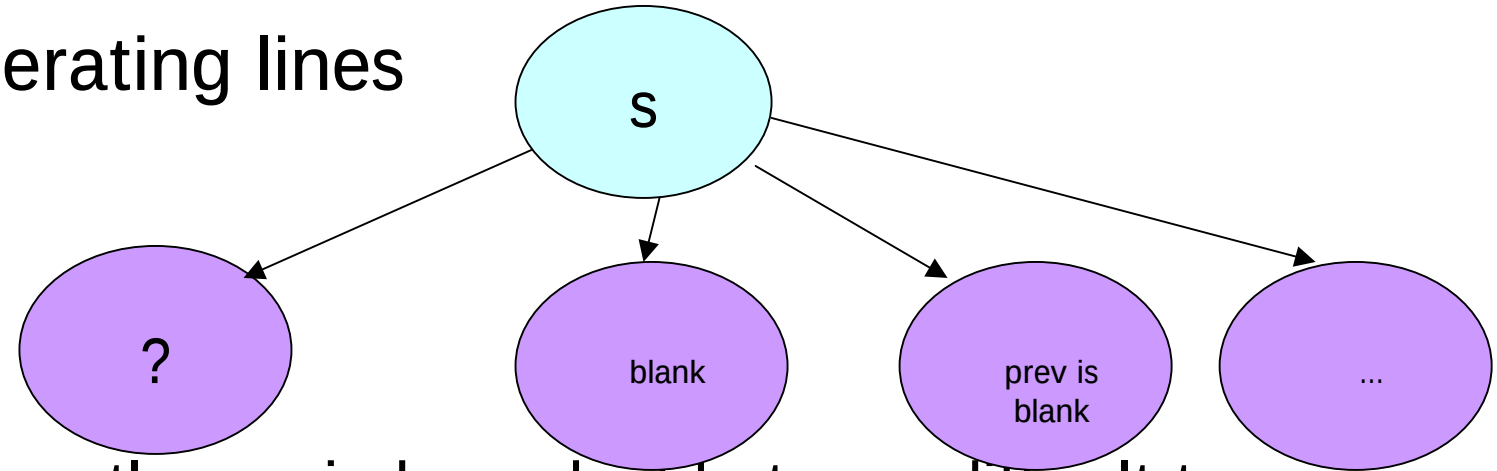
- Task: Given an FAQ document – a sequence of lines of text, classify each line as head, question, answer, or tail.
- Possible HMM model: four states (head, question, answer,tail).
  - States emit every token separately (very sparse)
  - States emit features of lines that are less sparse and helpful for disambiguation

Features in Experiments

begins-with-number
begins-with-ordinal
begins-with-punctuation
begins-with-question-word
begins-with-subject
blank
contains-alphanum
contains-bracketed-number
contains-http
contains-non-space
contains-number
contains-pipe
contains-question-mark
contains-question-word
ends-with-question-mark
first-alpha-is-capitalized
indented
indented-1-to-4
indented-5-to-10
more-than-one-third-space
only-punctuation
prev-is-blank
prev-begins-with-ordinal
shorter-than-30

Difficulties with Generative Models: Generating Multiple Features

- Generating lines



- Apparently non-independent but very difficult to model otherwise
- Easy with a conditional log-linear model – reverse the arcs

$$P(s'|s,line) = \frac{\exp(\sum_{j=1}^{m} f_j(s,line,s')\lambda_j)}{\sum_{k=1}^{K}\exp(\sum_{j=1}^{m} f_j(s,line,s_k)\lambda_j)}$$

Models Tested

- **ME-Stateless**: A single maximum entropy classifier applied to each line independently.
- **TokenHMM**: A fully-connected HMM with four states, one for each of the line categories, each of which generates individual tokens (groups of alphanumeric characters and individual punctuation characters).
- **FeatureHMM**: Identical to TokenHMM, only the lines in a document are first converted to sequences of features.
- **MEMM**: maximum entropy Markov model

Results

Learner	*Segmentation precision*	*Segmentation recall*
ME-Stateless	0.038	0.362
TokenHMM	0.276	0.140
FeatureHMM	0.413	0.529
MEMM	0.867	0.681

What is Next

- Theory - General ML perspective on Generative and Discriminative Models
- Examples – Traffic Lights and Word Sense Disambiguation
- The case of Part-of-Speech Tagging

The Classification Problem

Given a training set of iid samples $T=\{(X_1,Y_1) \ldots (X_n,Y_n)\}$ of input and class variables from an unknown distribution D(X,Y), estimate a function $\hat{h}(X)$ that predicts the class from the input variables

The goal is to come up with a hypothesis $\hat{h}(X)$ with minimum expected loss (usually 0-1 loss)

$$err(\hat{h}) = \sum_{<X,Y>\in\Omega} D(X,Y)\delta(Y \neq \hat{h}(X))$$

Under 0-1 loss the hypothesis with minimum expected loss is the Bayes optimal classifier

$$h(X) = \arg\max_{Y\in Y} D(Y \mid X)$$

Approaches to Solving Classification Problems - I

1. Generative. Try to estimate the probability distribution of the data D(X,Y)
   - specify a parametric model family $\{P_\theta(X,Y) : \theta \in \Theta\}$
   - choose parameters $\hat{\theta}$ by maximum likelihood on training data

$$L(T \mid \theta) = \prod_{i=1}^{n} P_\theta(X_i, Y_i)$$

   - estimate conditional probabilities by Bayes rule

$$P_{\hat{\theta}}(Y \mid X) = P_{\hat{\theta}}(X,Y) / P_{\hat{\theta}}(X)$$

   - classify new instances to the most probable class Y according to $P_{\hat{\theta}}(Y \mid X)$

Approaches to Solving Classification Problems - I

2. Discriminative. Try to estimate the conditional distribution D(Y|X) from data.
   - specify a parametric model family $\{P_\theta(Y \mid X) : \theta \in \Theta\}$
   - estimate parameters $\hat{\theta}$ by maximum conditional likelihood of training data $CL(T \mid \theta, X) = \prod_{i=1}^{n} P_\theta(Y_i \mid X_i)$
   - classify new instances to the most probable class Y according to $P_{\hat{\theta}}(Y \mid X)$
3. Discriminative. Distribution-free. Try to estimate $\hat{h}(X)$ directly from data so that its expected loss will be minimized

Axes for comparison of different approaches

- Asymptotic accuracy
- Accuracy for limited training data
- Speed of convergence to the best hypothesis
- Complexity of training
- Modeling ease

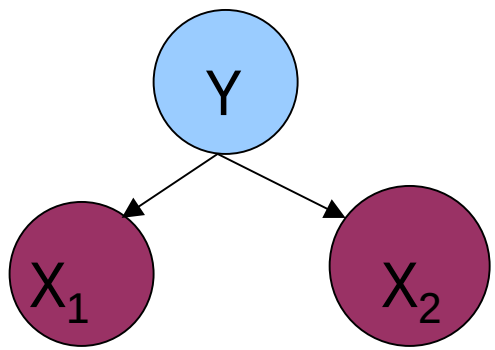
Generative-Discriminative Pairs

Definition: If a generative and discriminative parametric model family can represent the same set of conditional probability distributions $P_\theta(Y \mid X)$ they are a generative-discriminative pair

Example: Naïve Bayes and Logistic Regression

$Y \in \{1,2,...,K\}$ $\quad X_1, X_2 \in \{0,1\}$

$$P_{NB}(Y=i \mid X_1, X_2) = \frac{P(Y=i)P(X_1 \mid Y=i)P(X_2 \mid Y=i)}{\sum_{\bar{i}=1...K} P(Y=\bar{i})P(X_1 \mid Y=\bar{i})P(X_2 \mid Y=\bar{i})}$$



$$P_{LR}(Y=i \mid X_1, X_2) = \frac{\exp(\lambda_{i1}X_1 + \lambda_{i2}X_2 + \lambda_{i0})}{\sum_{\bar{i}=1...K} \exp(\lambda_{\bar{i}1}X_1 + \lambda_{\bar{i}2}X_2 + \lambda_{\bar{i}0})}$$

Comparison of Naïve Bayes and Logistic Regression

- The NB assumption that features are independent given the class is not made by logistic regression

$$P_{NB}(X_1, X_2 \mid Y = i) = P(X_1 \mid Y = i) P(X_2 \mid Y = i)$$

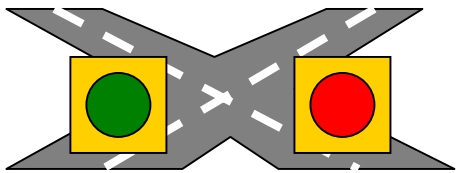
$$P_{LR}(X_1, X_2 \mid Y = i) = \frac{P(X_1, X_2)}{P(Y = i) \sum_{\bar{i}=1\ldots K} \exp(\lambda_{\bar{i}1} X_1 + \lambda_{\bar{i}2} X_2 + \lambda_{\bar{i}0})} \exp(\lambda_{i1} X_1 + \lambda_{i2} X_2 + \lambda_{i0})$$

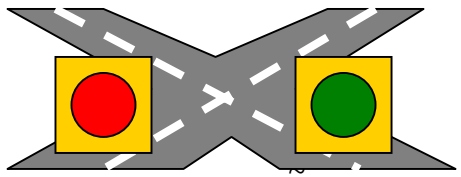
- The logistic regression model is more general because it allows a larger class of probability distributions for the features given classes

Example: Traffic Lights

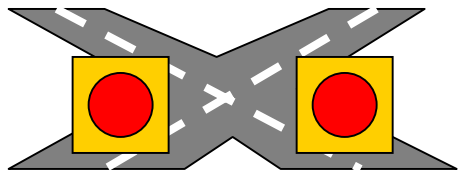
Reality

Lights Working





Lights Broken

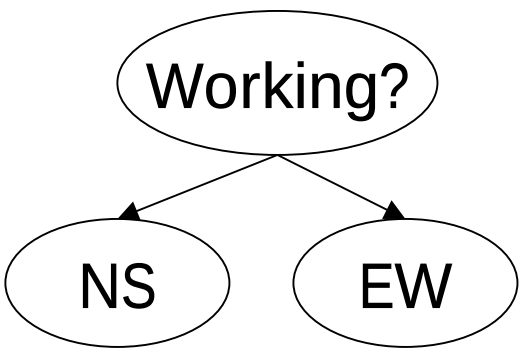


P(g,r,w) = 3/7

P(r,g,w) = 3/7

P(r,r,b) = 1/7

NB Model



- Model assumptions false!
- JL and CL estimates differ…

JL: P(w) = 6/7 CL: $\tilde{P}(w) = \varepsilon$

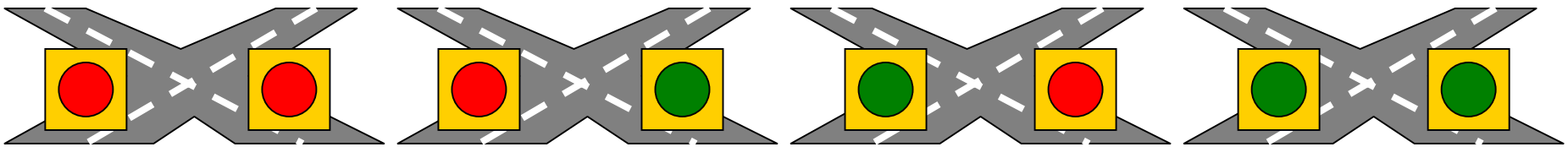
P(r|w) = ½ $\tilde{P}(r|w) = ½$

P(r|b) = 1 $\tilde{P}(r|b) = 1$

Joint Traffic Lights

Lights Working

3/14 3/14 3/14 3/14



2/14

0 0 0

Lights Broken

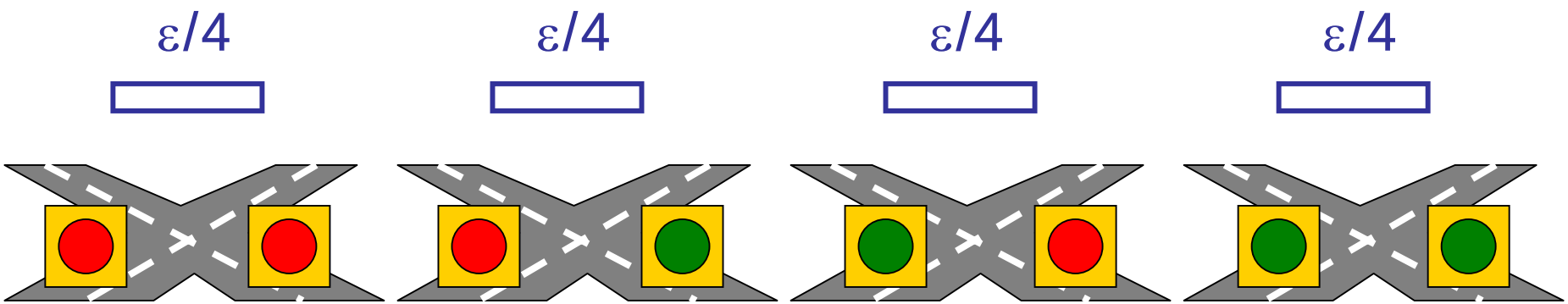
Conditional likelihood of working is > ½!
Incorrectly assigned!

→ Accuracy: 6/7

Conditional likelihood of working is 1

Conditional Traffic Lights

Lights Working



0 0 0

Conditional likelihood of working is still 1

1-ε

Lights Broken

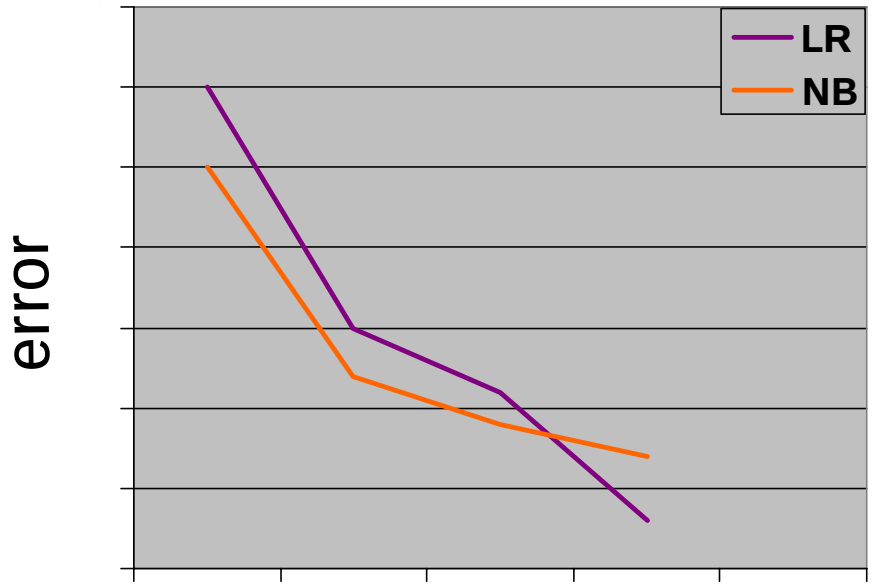
Now correctly assigned to broken.

Accuracy: 7/7
CL perfect (1)
JL low (to 0)

Comparison of Naïve Bayes and Logistic Regression

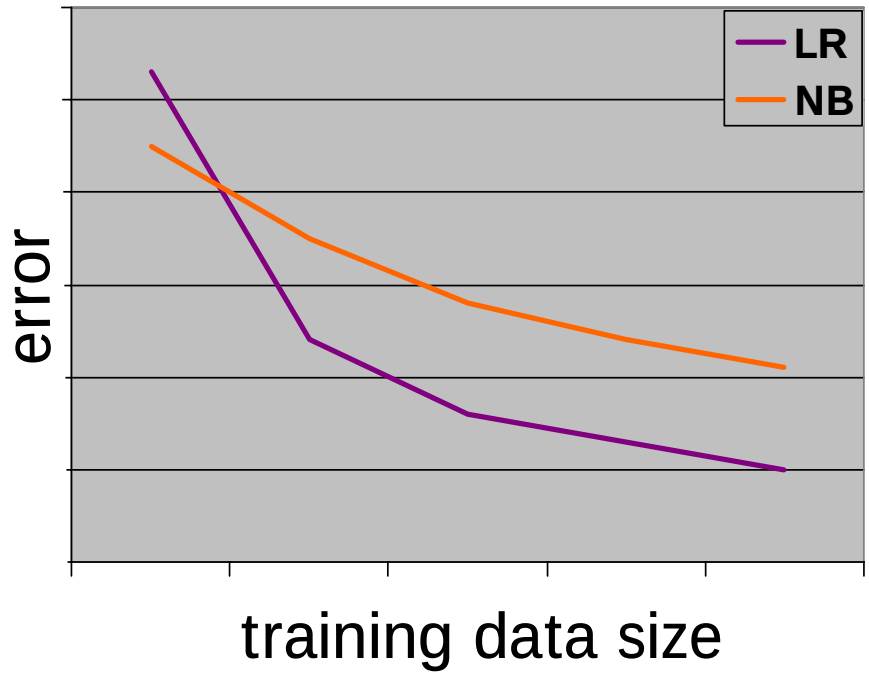
	Naïve Bayes	Logistic Regression
Accuracy		+
Convergence	+	
Training Speed	+	
Model assumptions	independence of features given class	Linear log-odds $\log(\frac{P(X_1, X_2 \mid Y = i)}{P(X_1, X_2 \mid Y = j)})$
Advantages	Faster convergence, uses information in P(X), faster training	More robust and accurate because fewer assumptions
Disadvantages	Large bias if the independence assumptions are very wrong	Harder parameter estimation problem, ignores information in P(X)

Some Experimental Comparisons



Ng & Jordan 2002

(15 datasets from UCI ML)



Klein & Manning 2002

(WSD *line* and *hard* data)

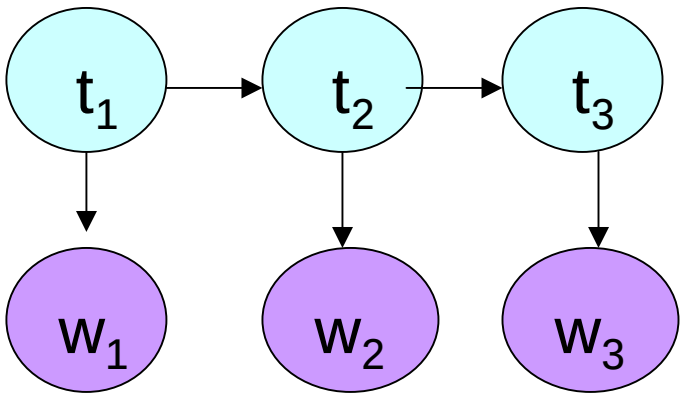
Part-of-Speech Tagging Useful Features

- In most cases information about surrounding words/tags is strong disambiguator

  *"The long **fenestration** was tiring . "*

- Useful features
  - tags of previous/following words
    P(NN|JJ)=.45;P(VBP|JJ)=0.0005
  - identity of word being tagged/surrounding words
  - suffix/prefix for unknown words, hyphenation, capitalization
  - longer distance features
  - others we haven't figured out yet

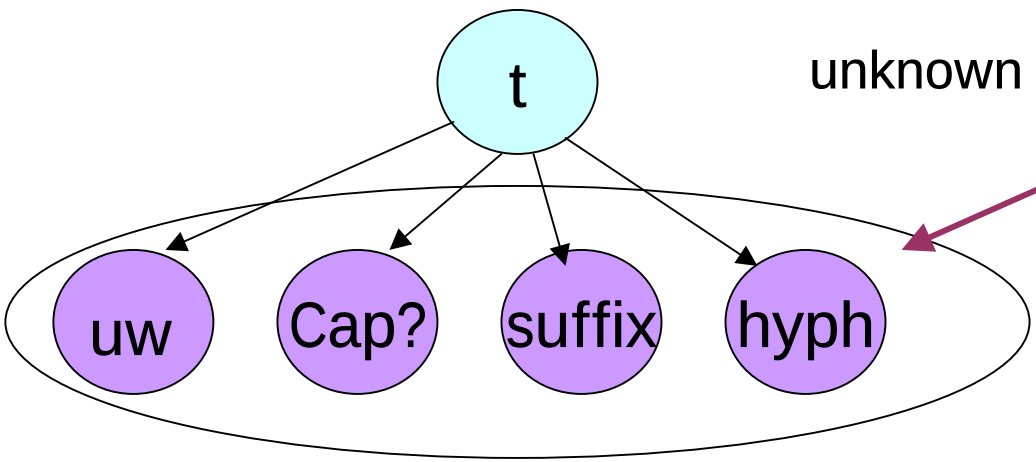
HMM Tagging Models - I



Independence Assumptions

- t_i is independent of $t_1 \ldots t_{i-2}$ and $w_1 \ldots w_{i-1}$ given t_{i-1}
- words are independent given their tags

states can be single tags or pairs of successive tags or variable length sequences of last tags

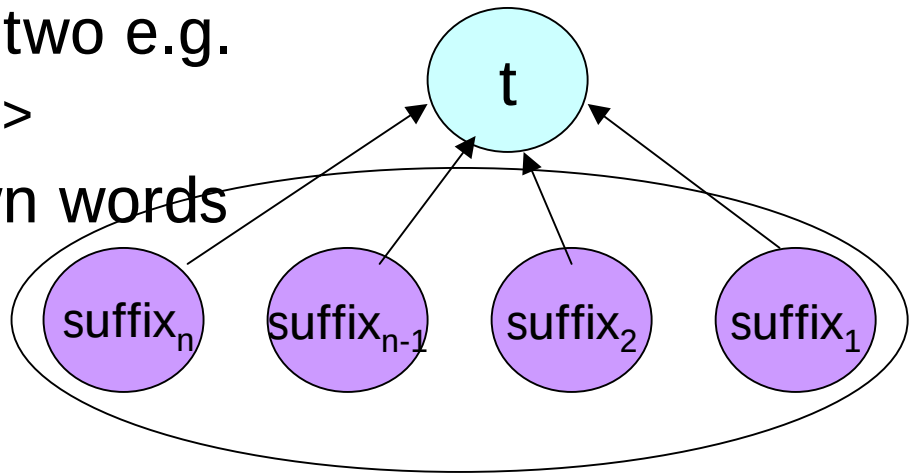


unknown words (Weischedel et al. 93)

HMM Tagging Models - Brants 2000

- Highly competitive with other state-of-the art models
- Trigram HMM with smoothed transition probabilities
- Capitalization feature becomes part of the state – each tag state is split into two e.g.

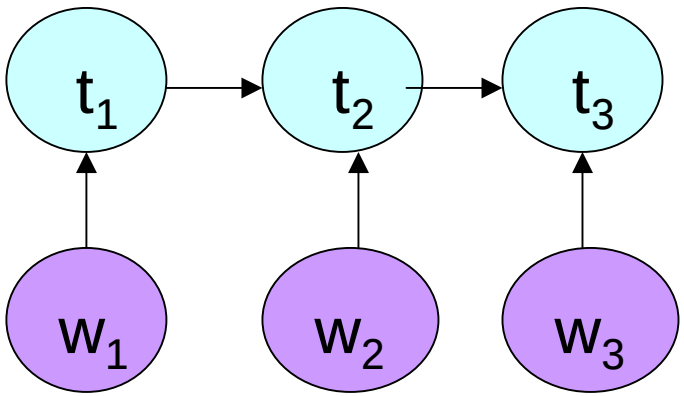
  NN $\rightarrow$ <NN,cap>,<NN,not cap>
- Suffix features for unknown words



$$P(w \mid tag) = P(suffix \mid tag)(w \mid suffix)$$
$$\approx \hat{P}(suffix)\tilde{P}(tag \mid suffix) / \hat{P}(tag)$$

$$\tilde{P}(tag \mid suffix_n) = \lambda_1 \hat{P}(tag \mid suffix_n) + \lambda_2 \hat{P}(tag \mid suffix_{n-1}) + \ldots + \lambda_n \hat{P}(tag)$$

CMM Tagging Models



Independence Assumptions

- t_i is independent of $t_1 \ldots t_{i-2}$ and $w_1 \ldots w_{i-1}$ given t_{i-1}
- t_i is independent of all following observations
- no independence assumptions on the observation sequence

•Dependence of current tag on previous and future observations can be added; overlapping features of the observation can be taken as predictors

MEMM Tagging Models -II

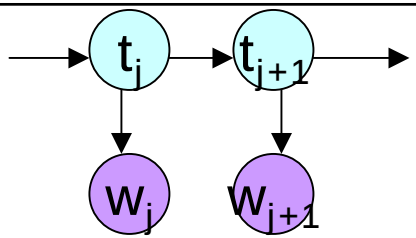
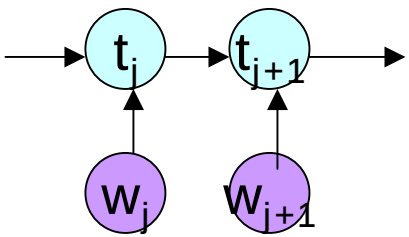
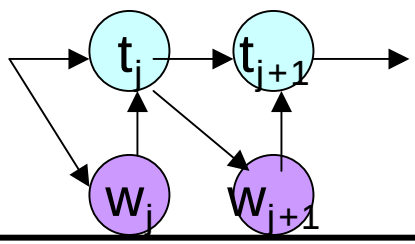
Ratnaparkhi (1996)

- local distributions are estimated using maximum entropy models
- used previous two tags, current word, previous two words, next two words
- suffix, prefix, hyphenation, and capitalization features for unknown words

Model	Overall Accuracy	Unknown Words
HMM (Brants 2000)	96.7	85.5
MEMM(Ratn 1996)	96.63	85.56
MEMM(T. et al 2003)	97	88

HMM vs CMM – I

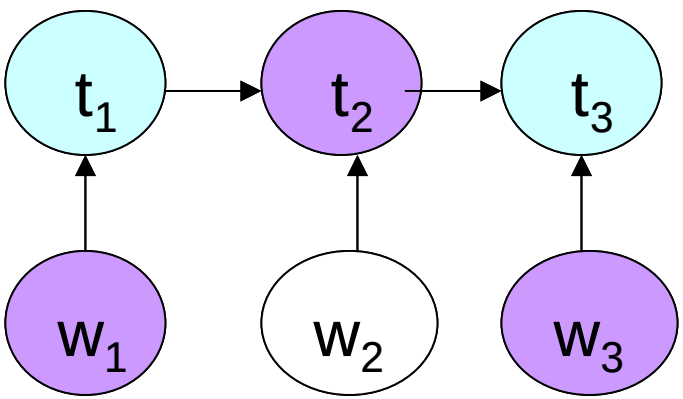
Johnson (2001)

Model	Accuracy
	95.5%
	94.4%
	95.3%

HMM vs CMM - II

- The per-state conditioning of the CMM has been observed to exhibit *label bias* (Bottou, Lafferty) and *observation bias* (Klein & Manning)

HMM	CMM	CMM+
91.23	89.22	90.44



Unobserving words with unambiguous tags improved performance significantly

Summary of Tagging Review

For tagging, the change from generative to discriminative model **does not by itself** result in great improvement

One profits from discriminative models for specifying dependence on **overlapping features of the observation** such as spelling, suffix analysis,etc

The CMM model allows integration of rich features of the observations, but suffers strongly from assuming independence from following observations; this effect can be relieved by adding dependence on following words

This additional power (of the CMM ,CRF, Perceptron models) has been shown to result in improvements in accuracy

The **higher accuracy** of discriminative models comes at the price of **much slower training**