

6.864 (Fall 2006): Lecture 6 Log-Linear Models

Michael Collins, MIT

Trigram Models

- Estimate a distribution $P(w_i|w_1, w_2, \dots, w_{i-1})$ given previous “history” $w_1, \dots, w_{i-1} =$

Third, the notion “grammatical in English” cannot be identified in any way with the notion “high order of statistical approximation to English”. It is fair to assume that neither sentence (1) nor (2) (nor indeed any part of these sentences) has ever occurred in an English discourse. Hence, in any statistical

- **Trigram estimates:**

$$P(\text{model}|w_1, \dots, w_{i-1}) = \lambda_1 P_{ML}(\text{model}|w_{i-2} = \text{any}, w_{i-1} = \text{statistical}) + \lambda_2 P_{ML}(\text{model}|w_{i-1} = \text{statistical}) + \lambda_3 P_{ML}(\text{model})$$

$$\text{where } \lambda_i \geq 0, \sum_i \lambda_i = 1, P_{ML}(y|x) = \frac{\text{Count}(x,y)}{\text{Count}(x)}$$

The Language Modeling Problem

- w_i is the i 'th word in a document
- Estimate a distribution $P(w_i|w_1, w_2, \dots, w_{i-1})$ given previous “history” w_1, \dots, w_{i-1} .
- E.g., $w_1, \dots, w_{i-1} =$

Third, the notion “grammatical in English” cannot be identified in any way with the notion “high order of statistical approximation to English”. It is fair to assume that neither sentence (1) nor (2) (nor indeed any part of these sentences) has ever occurred in an English discourse. Hence, in any statistical

Trigram Models

$$P(\text{model}|w_1, \dots, w_{i-1}) = \lambda_1 P_{ML}(\text{model}|w_{i-2} = \text{any}, w_{i-1} = \text{statistical}) + \lambda_2 P_{ML}(\text{model}|w_{i-1} = \text{statistical}) + \lambda_3 P_{ML}(\text{model})$$

- Makes use of only bigram, trigram, unigram estimates
- Many other “features” of w_1, \dots, w_{i-1} may be useful, e.g.:

$$P_{ML}(\text{model} \mid w_{i-2} = \text{any})$$

$$P_{ML}(\text{model} \mid w_{i-1} \text{ is an adjective})$$

$$P_{ML}(\text{model} \mid w_{i-1} \text{ ends in “ical”})$$

$$P_{ML}(\text{model} \mid \text{author} = \text{Chomsky})$$

$$P_{ML}(\text{model} \mid \text{“model” does not occur somewhere in } w_1, \dots, w_{i-1})$$

$$P_{ML}(\text{model} \mid \text{“grammatical” occurs somewhere in } w_1, \dots, w_{i-1})$$

A Naive Approach

$$\begin{aligned} P(\text{model}|w_1, \dots, w_{i-1}) = & \\ \lambda_1 P_{ML}(\text{model}|w_{i-2} = \text{any}, w_{i-1} = \text{statistical}) + & \\ \lambda_2 P_{ML}(\text{model}|w_{i-1} = \text{statistical}) + & \\ \lambda_3 P_{ML}(\text{model}) + & \\ \lambda_4 P_{ML}(\text{model}|w_{i-2} = \text{any}) + & \\ \lambda_5 P_{ML}(\text{model}|w_{i-1} \text{ is an adjective}) + & \\ \lambda_6 P_{ML}(\text{model}|w_{i-1} \text{ ends in "ical"}) + & \\ \lambda_7 P_{ML}(\text{model}|author = \text{Chomsky}) + & \\ \lambda_8 P_{ML}(\text{model}|\text{"model" does not occur somewhere in } w_1, \dots, w_{i-1}) + & \\ \lambda_9 P_{ML}(\text{model}|\text{"grammatical" occurs somewhere in } w_1, \dots, w_{i-1}) & \end{aligned}$$

This quickly becomes very unwieldy...

A Second Example: Part-of-Speech Tagging

Hispaniola/**NNP** quickly/**RB** became/**VB** an/**DT** important/**JJ** base/**??** from which Spain expanded its empire into the rest of the Western Hemisphere .

- There are many possible tags in the position **??**
{**NN, NNS, Vt, Vi, IN, DT, ...**}

- The task: model the distribution

$$P(t_i|t_1, \dots, t_{i-1}, w_1 \dots w_n)$$

where t_i is the i 'th tag in the sequence, w_i is the i 'th word

A Second Example: Part-of-Speech Tagging

INPUT:

Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results.

OUTPUT:

Profits/**N** soared/**V** at/**P** Boeing/**N** Co./**N** ,/, easily/**ADV** topping/**V** forecasts/**N** on/**P** Wall/**N** Street/**N** ,/, as/**P** their/**POSS** CEO/**N** Alan/**N** Mulally/**N** announced/**V** first/**ADJ** quarter/**N** results/**N** ./.

N = Noun
V = Verb
P = Preposition
Adv = Adverb
Adj = Adjective
...

A Second Example: Part-of-Speech Tagging

Hispaniola/**NNP** quickly/**RB** became/**VB** an/**DT** important/**JJ** base/**??** from which Spain expanded its empire into the rest of the Western Hemisphere .

- The task: model the distribution

$$P(t_i|t_1, \dots, t_{i-1}, w_1 \dots w_n)$$

where t_i is the i 'th tag in the sequence, w_i is the i 'th word

- Again: many “features” of $t_1, \dots, t_{i-1}, w_1 \dots w_n$ may be relevant

$$\begin{aligned} P_{ML}(\text{NN} \mid w_i = \text{base}) & \\ P_{ML}(\text{NN} \mid t_{i-1} \text{ is JJ}) & \\ P_{ML}(\text{NN} \mid w_i \text{ ends in "e"}) & \\ P_{ML}(\text{NN} \mid w_i \text{ ends in "se"}) & \\ P_{ML}(\text{NN} \mid w_{i-1} \text{ is "important"}) & \\ P_{ML}(\text{NN} \mid w_{i+1} \text{ is "from"}) & \end{aligned}$$

Overview

- Log-linear models
- The maximum-entropy property
- Smoothing, feature selection etc. in log-linear models

Language Modeling

- x is a “history” w_1, w_2, \dots, w_{i-1} , e.g.,
Third, the notion “grammatical in English” cannot be identified in any way with the notion “high order of statistical approximation to English”. It is fair to assume that neither sentence (1) nor (2) (nor indeed any part of these sentences) has ever occurred in an English discourse. Hence, in any statistical
- y is an “outcome” w_i

The General Problem

- We have some **input domain** \mathcal{X}
- Have a finite **label set** \mathcal{Y}
- Aim is to provide a **conditional probability** $P(y | x)$ for any x, y where $x \in \mathcal{X}, y \in \mathcal{Y}$

Feature Vector Representations

- Aim is to provide a conditional probability $P(y | x)$ for “decision” y given “history” x
- A **feature** is a function $f(x, y) \in \mathbb{R}$
(Often **binary features** or **indicator functions** $f(x, y) \in \{0, 1\}$).
- Say we have m features ϕ_k for $k = 1 \dots m$
 \Rightarrow A **feature vector** $\vec{\phi}(x, y) \in \mathbb{R}^m$ for any x, y

Language Modeling

- x is a “history” w_1, w_2, \dots, w_{i-1} , e.g.,

Third, the notion “grammatical in English” cannot be identified in any way with the notion “high order of statistical approximation to English”. It is fair to assume that neither sentence (1) nor (2) (nor indeed any part of these sentences) has ever occurred in an English discourse. Hence, in any statistical

- y is an “outcome” w_i

$$\phi_7(x, y) = \begin{cases} 1 & \text{if } y = \text{model, author} = \text{Chomsky} \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_8(x, y) = \begin{cases} 1 & \text{if } y = \text{model, “model” is not in } w_1, \dots, w_{i-1} \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_9(x, y) = \begin{cases} 1 & \text{if } y = \text{model, “grammatical” is in } w_1, \dots, w_{i-1} \\ 0 & \text{otherwise} \end{cases}$$

- Example features:

$$\phi_1(x, y) = \begin{cases} 1 & \text{if } y = \text{model} \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_2(x, y) = \begin{cases} 1 & \text{if } y = \text{model and } w_{i-1} = \text{statistical} \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_3(x, y) = \begin{cases} 1 & \text{if } y = \text{model, } w_{i-2} = \text{any, } w_{i-1} = \text{statistical} \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_4(x, y) = \begin{cases} 1 & \text{if } y = \text{model, } w_{i-2} = \text{any} \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_5(x, y) = \begin{cases} 1 & \text{if } y = \text{model, } w_{i-1} \text{ is an adjective} \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_6(x, y) = \begin{cases} 1 & \text{if } y = \text{model, } w_{i-1} \text{ ends in “ical”} \\ 0 & \text{otherwise} \end{cases}$$

Defining Features in Practice

- We had the following “trigram” feature:

$$\phi_3(x, y) = \begin{cases} 1 & \text{if } y = \text{model, } w_{i-2} = \text{any, } w_{i-1} = \text{statistical} \\ 0 & \text{otherwise} \end{cases}$$

- In practice, we would probably introduce one trigram feature for every trigram seen in the training data: i.e., for all trigrams (u, v, w) seen in training data, create a feature

$$\phi_{N(u,v,w)}(x, y) = \begin{cases} 1 & \text{if } y = w, w_{i-2} = u, w_{i-1} = v \\ 0 & \text{otherwise} \end{cases}$$

where $N(u, v, w)$ is a function that maps each (u, v, w) trigram to a different integer

The POS-Tagging Example

- Each x is a “history” of the form $\langle t_1, t_2, \dots, t_{i-1}, w_1 \dots w_n, i \rangle$
- Each y is a POS tag, such as $NN, NNS, Vt, Vi, IN, DT, \dots$
- We have m features $\phi_k(x, y)$ for $k = 1 \dots m$

For example:

$$\phi_1(x, y) = \begin{cases} 1 & \text{if current word } w_i \text{ is base and } y = Vt \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_2(x, y) = \begin{cases} 1 & \text{if current word } w_i \text{ ends in ing and } y = VBG \\ 0 & \text{otherwise} \end{cases}$$

...

The Full Set of Features in [Ratnaparkhi 96]

- Contextual Features, e.g.,

$$\phi_{103}(x, y) = \begin{cases} 1 & \text{if } \langle t_{i-2}, t_{i-1}, y \rangle = \langle DT, JJ, Vt \rangle \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_{104}(x, y) = \begin{cases} 1 & \text{if } \langle t_{i-1}, y \rangle = \langle JJ, Vt \rangle \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_{105}(x, y) = \begin{cases} 1 & \text{if } \langle y \rangle = \langle Vt \rangle \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_{106}(x, y) = \begin{cases} 1 & \text{if previous word } w_{i-1} = \textit{the} \text{ and } y = Vt \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_{107}(x, y) = \begin{cases} 1 & \text{if next word } w_{i+1} = \textit{the} \text{ and } y = Vt \\ 0 & \text{otherwise} \end{cases}$$

The Full Set of Features in [Ratnaparkhi 96]

- Word/tag features for all word/tag pairs, e.g.,

$$\phi_{100}(x, y) = \begin{cases} 1 & \text{if current word } w_i \text{ is base and } y = Vt \\ 0 & \text{otherwise} \end{cases}$$

- Spelling features for all prefixes/suffixes of length ≤ 4 , e.g.,

$$\phi_{101}(x, y) = \begin{cases} 1 & \text{if current word } w_i \text{ ends in ing and } y = VBG \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_{102}(h, t) = \begin{cases} 1 & \text{if current word } w_i \text{ starts with pre and } y = NN \\ 0 & \text{otherwise} \end{cases}$$

The Final Result

- We can come up with practically any questions (*features*) regarding history/tag pairs.
- For a given history $x \in \mathcal{X}$, each label in \mathcal{Y} is mapped to a different feature vector

$$\begin{aligned} \phi(\langle JJ, DT, \langle \text{Hispaniola}, \dots \rangle, 6 \rangle, Vt) &= 1001011001001100110 \\ \phi(\langle JJ, DT, \langle \text{Hispaniola}, \dots \rangle, 6 \rangle, JJ) &= 0110010101011110010 \\ \phi(\langle JJ, DT, \langle \text{Hispaniola}, \dots \rangle, 6 \rangle, NN) &= 0001111101001100100 \\ \phi(\langle JJ, DT, \langle \text{Hispaniola}, \dots \rangle, 6 \rangle, IN) &= 0001011011000000010 \\ &\dots \end{aligned}$$

Parameter Vectors

- Given features $\phi_k(x, y)$ for $k = 1 \dots m$, also define a **parameter vector** $\mathbf{W} \in \mathbb{R}^m$
- Each (x, y) pair is then mapped to a “score”

$$\sum_k \mathbf{W}_k \phi_k(x, y)$$

Log-Linear Models

- We have some input domain \mathcal{X} , and a finite label set \mathcal{Y} . Aim is to provide a conditional probability $P(y | x)$ for any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.
- A feature is a function $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$
(Often binary features or indicator functions $f : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$).
- Say we have m features ϕ_k for $k = 1 \dots m$
 \Rightarrow A feature vector $\phi(x, y) \in \mathbb{R}^m$ for any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.
- We also have a **parameter vector** $\mathbf{W} \in \mathbb{R}^m$

Language Modeling

- x is a “history” w_1, w_2, \dots, w_{i-1} , e.g.,
Third, the notion “grammatical in English” cannot be identified in any way with the notion “high order of statistical approximation to English”. It is fair to assume that neither sentence (1) nor (2) (nor indeed any part of these sentences) has ever occurred in an English discourse. Hence, in any statistical
- Each possible y gets a different score:

$$\begin{aligned} \sum_k \mathbf{W}_k \phi_k(x, model) &= 5.6 & \sum_k \mathbf{W}_k \phi_k(x, the) &= -3.2 \\ \sum_k \mathbf{W}_k \phi_k(x, is) &= 1.5 & \sum_k \mathbf{W}_k \phi_k(x, of) &= 1.3 \\ \sum_k \mathbf{W}_k \phi_k(x, models) &= 4.5 & \dots & \end{aligned}$$

- We define

$$P(y | x, \mathbf{W}) = \frac{e^{\sum_k \mathbf{W}_k \phi_k(x, y)}}{\sum_{y' \in \mathcal{Y}} e^{\sum_k \mathbf{W}_k \phi_k(x, y')}}$$

More About Log-Linear Models

- Why the name?

$$\log P(y | x, \mathbf{W}) = \underbrace{\mathbf{W} \cdot \phi(x, y)}_{\text{Linear term}} - \underbrace{\log \sum_{y' \in \mathcal{Y}} e^{\mathbf{W} \cdot \phi(x, y')}}_{\text{Normalization term}}$$

- Maximum-likelihood estimates given training sample (x_i, y_i) for $i = 1 \dots n$, each $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$:

$$\mathbf{W}_{ML} = \operatorname{argmax}_{\mathbf{W} \in \mathbb{R}^m} L(\mathbf{W})$$

where

$$\begin{aligned} L(\mathbf{W}) &= \sum_{i=1}^n \log P(y_i | x_i) \\ &= \sum_{i=1}^n \mathbf{W} \cdot \phi(x_i, y_i) - \sum_{i=1}^n \log \sum_{y' \in \mathcal{Y}} e^{\mathbf{W} \cdot \phi(x_i, y')} \end{aligned}$$

Gradient Ascent Methods

- Need to maximize $L(\mathbf{W})$ where

$$\left. \frac{dL}{d\mathbf{W}} \right|_{\mathbf{W}} = \sum_{i=1}^n \phi(x_i, y_i) - \sum_{i=1}^n \sum_{y' \in \mathcal{Y}} \phi(x_i, y') P(y' | x_i, \mathbf{W})$$

Initialization: $\mathbf{W} = 0$

Iterate until convergence:

- Calculate $\Delta = \left. \frac{dL}{d\mathbf{W}} \right|_{\mathbf{W}}$
- Calculate $\beta_* = \operatorname{argmax}_{\beta} L(\mathbf{W} + \beta \Delta)$ (**Line Search**)
- Set $\mathbf{W} \leftarrow \mathbf{W} + \beta_* \Delta$

Calculating the Maximum-Likelihood Estimates

- Need to maximize:

$$L(\mathbf{W}) = \sum_{i=1}^n \mathbf{W} \cdot \phi(x_i, y_i) - \sum_{i=1}^n \log \sum_{y' \in \mathcal{Y}} e^{\mathbf{W} \cdot \phi(x_i, y')}$$

- Calculating gradients:

$$\begin{aligned} \left. \frac{dL}{d\mathbf{W}} \right|_{\mathbf{W}} &= \sum_{i=1}^n \phi(x_i, y_i) - \sum_{i=1}^n \frac{\sum_{y' \in \mathcal{Y}} \phi(x_i, y') e^{\mathbf{W} \cdot \phi(x_i, y')}}{\sum_{z' \in \mathcal{Y}} e^{\mathbf{W} \cdot \phi(x_i, z')}} \\ &= \sum_{i=1}^n \phi(x_i, y_i) - \sum_{i=1}^n \sum_{y' \in \mathcal{Y}} \phi(x_i, y') \frac{e^{\mathbf{W} \cdot \phi(x_i, y')}}{\sum_{z' \in \mathcal{Y}} e^{\mathbf{W} \cdot \phi(x_i, z')}} \\ &= \underbrace{\sum_{i=1}^n \phi(x_i, y_i)}_{\text{Empirical counts}} - \underbrace{\sum_{i=1}^n \sum_{y' \in \mathcal{Y}} \phi(x_i, y') P(y' | x_i, \mathbf{W})}_{\text{Expected counts}} \end{aligned}$$

Conjugate Gradient Methods

- (Vanilla) gradient ascent can be very slow
- Conjugate gradient methods require calculation of gradient at each iteration, but do a line search in **a direction which is a function of the current gradient, and the previous step taken.**
- Conjugate gradient packages are widely available
In general: they require a function

$$\text{calc_gradient}(\mathbf{W}) \rightarrow \left(L(\mathbf{W}), \left. \frac{dL}{d\mathbf{W}} \right|_{\mathbf{W}} \right)$$

and that's about it!

Overview

- Log-linear models
- **The maximum-entropy property**
- Smoothing, feature selection etc. in log-linear models

Maximum-Entropy Properties of Log-Linear Models

- The **entropy** of any distribution is:

$$H(p) = - \left(\frac{1}{n} \sum_i \sum_{y \in \mathcal{Y}} p(y | x_i) \log p(y | x_i) \right)$$

- Entropy is a measure of “smoothness” of a distribution
- In this case, entropy is maximized by uniform distribution,

$$p(y | x_i) = \frac{1}{|\mathcal{Y}|} \text{ for all } y, x_i$$

Maximum-Entropy Properties of Log-Linear Models

- We define the set of distributions which satisfy linear constraints implied by the data:

$$\mathcal{P} = \left\{ p : \underbrace{\sum_i \phi(x_i, y_i)}_{\text{Empirical counts}} = \underbrace{\sum_i \sum_{y \in \mathcal{Y}} p(y | x_i) \phi(x_i, y)}_{\text{Expected counts}} \right\}$$

here, p is an $n \times |\mathcal{Y}|$ vector defining $P(y | x_i)$ for all i, y .

- Note that at least one distribution satisfies these constraints, i.e.,

$$p(y | x_i) = \begin{cases} 1 & \text{if } y = y_i \\ 0 & \text{otherwise} \end{cases}$$

The Maximum-Entropy Solution

- The maximum entropy model is

$$p_* = \operatorname{argmax}_{p \in \mathcal{P}} H(p)$$

- Intuition: find a distribution which
 1. satisfies the constraints
 2. is as smooth as possible

Maximum-Entropy Properties of Log-Linear Models

- Consider the distribution

$$P(y|x, \mathbf{W}^*)$$

defined by the maximum-likelihood estimates $\mathbf{W}^* = \arg \max L(\mathbf{W})$

- Then $P(y|x, \mathbf{W}^*)$ is the maximum-entropy distribution

Overview

- Log-linear models
- The maximum-entropy property
- Smoothing, feature selection etc. in log-linear models

Is the Maximum-Entropy Property Useful?

- Intuition: find a distribution which
 1. satisfies the constraints
 2. is as smooth as possible
- One problem: the constraints are define by *empirical counts* from the data.
- Another problem: no formal relationship between maximum-entropy property and generalization(?) (at least none is given in the NLP literature)

Smoothing in Maximum Entropy Models

- Say we have a feature:

$$\phi_{100}(h, t) = \begin{cases} 1 & \text{if current word } w_i \text{ is base and } t = \text{Vt} \\ 0 & \text{otherwise} \end{cases}$$

- In training data, base is seen 3 times, with Vt every time
- Maximum likelihood solution satisfies

$$\sum_i \phi_{100}(x_i, y_i) = \sum_i \sum_y p(y | x_i, \mathbf{W}) \phi_{100}(x_i, y)$$

- $\Rightarrow p(\text{Vt} | x_i, \mathbf{W}) = 1$ for any history x_i where $w_i = \text{base}$
- $\Rightarrow \mathbf{W}_{100} \rightarrow \infty$ at maximum-likelihood solution (most likely)
- $\Rightarrow p(\text{Vt} | x, \mathbf{W}) = 1$ for any test data history x where $w = \text{base}$

A Simple Approach: Count Cut-Offs

- [Ratnaparkhi 1998] (PhD thesis): include all features that occur 5 times or more in training data. i.e.,

$$\sum_i \phi_k(x_i, y_i) \geq 5$$

for all features ϕ_k .

The Bayesian Justification for Gaussian Priors

- In *Bayesian* methods, combine the log-likelihood $P(\text{data} \mid \mathbf{W})$ with a prior over parameters, $P(\mathbf{W})$

$$P(\mathbf{W} \mid \text{data}) = \frac{P(\text{data} \mid \mathbf{W})P(\mathbf{W})}{\int_{\mathbf{W}} P(\text{data} \mid \mathbf{W})P(\mathbf{W})d\mathbf{W}}$$

- The **MAP** (Maximum A-Posteriori) estimates are

$$\begin{aligned} \mathbf{W}_{MAP} &= \operatorname{argmax}_{\mathbf{W}} P(\mathbf{W} \mid \text{data}) \\ &= \operatorname{argmax}_{\mathbf{W}} \left(\underbrace{\log P(\text{data} \mid \mathbf{W})}_{\text{Log-Likelihood}} + \underbrace{\log P(\mathbf{W})}_{\text{Prior}} \right) \end{aligned}$$

- Gaussian prior: $P(\mathbf{W}) \propto e^{-\sum_k \mathbf{W}_k^2 / 2\sigma^2}$
 $\Rightarrow \log P(\mathbf{W}) = -\sum_k \mathbf{W}_k^2 / 2\sigma^2 + C$

Gaussian Priors

- Modified loss function

$$L(\mathbf{W}) = \sum_{i=1}^n \mathbf{W} \cdot \phi(x_i, y_i) - \sum_{i=1}^n \log \sum_{y' \in \mathcal{Y}} e^{\mathbf{W} \cdot \phi(x_i, y')} - \sum_{k=1}^m \frac{\mathbf{W}_k^2}{2\sigma^2}$$

- Calculating gradients:

$$\frac{dL}{d\mathbf{W}} \Big|_{\mathbf{W}} = \underbrace{\sum_{i=1}^n \phi(x_i, y_i)}_{\text{Empirical counts}} - \underbrace{\sum_{i=1}^n \sum_{y' \in \mathcal{Y}} \phi(x_i, y') P(y' \mid x_i, \mathbf{W})}_{\text{Expected counts}} - \frac{1}{\sigma^2} \mathbf{W}$$

- Can run conjugate gradient methods as before

- Adds a penalty for large weights

Experiments with Gaussian Priors

- [Chen and Rosenfeld, 1998]: apply maximum entropy models to language modeling: Estimate $P(w_i \mid w_{i-2}, w_{i-1})$

- Unigram, bigram, trigram features, e.g.,

$$\begin{aligned} \phi_1(w_{i-2}, w_{i-1}, w_i) &= \begin{cases} 1 & \text{if trigram is (the , dog , laughs)} \\ 0 & \text{otherwise} \end{cases} \\ \phi_2(w_{i-2}, w_{i-1}, w_i) &= \begin{cases} 1 & \text{if bigram is (dog , laughs)} \\ 0 & \text{otherwise} \end{cases} \\ \phi_3(w_{i-2}, w_{i-1}, w_i) &= \begin{cases} 1 & \text{if unigram is (laughs)} \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

$$P(w_i \mid w_{i-2}, w_{i-1}) = \frac{e^{\sum_k \phi_k(w_{i-2}, w_{i-1}, w_i) \cdot \mathbf{W}}}{\sum_w e^{\sum_k \phi_k(w_{i-2}, w_{i-1}, w) \cdot \mathbf{W}}}$$

Experiments with Gaussian Priors

- In regular (unsmoothed) maxent, if all n-gram features are included, then it's equivalent to maximum-likelihood estimates!

$$P(w_i | w_{i-2}, w_{i-1}) = \frac{\text{Count}(w_{i-2}, w_{i-1}, w_i)}{\text{Count}(w_{i-2}, w_{i-1})}$$

- [Chen and Rosenfeld, 1998]: with gaussian priors, get very good results. Performs as well as or better than standardly used “discounting methods” (see lecture 2).
- Note: their method uses development set to optimize σ parameters
- Downside: computing $\sum_w e^{\sum_k \phi_k(w_{i-2}, w_{i-1}, w) \cdot \mathbf{W}}$ is **SLOW**.

Figures from [Ratnaparkhi 1998] (PhD thesis)

- The task: PP attachment ambiguity
- **ME Default:** Count cut-off of 5
- **ME Tuned:** Count cut-offs vary for 4-tuples, 3-tuples, 2-tuples, unigram features
- **ME IFS:** feature selection method

Feature Selection Methods

- Goal: find *a small number of features* which make good progress in optimizing log-likelihood
- A greedy method:

Step 1 Throughout the algorithm, maintain a set of active features. Initialize this set to be empty.

Step 2 Choose a feature from outside of the set of active features which has largest estimated impact in terms of increasing the log-likelihood and add this to the active feature set.

Step 3 Minimize $L(\mathbf{W})$ with respect to the set of active features. Return to **Step 2**.

Experiment	Accuracy	Training Time	# of Features
ME Default	82.0%	10 min	4028
ME Tuned	83.7%	10 min	83875
ME IFS	80.5%	30 hours	387
DT Default	72.2%	1 min	
DT Tuned	80.4%	10 min	
DT Binary	-	1 week +	
Baseline	70.4%		

Table 8.2: Maximum Entropy (ME) and Decision Tree (DT) Experiments on PP attachment

Figures from [Ratnaparkhi 1998] (PhD thesis)

- A second task: text classification, identifying articles about acquisitions

Summary

- Introduced log-linear models as general approach for modeling conditional probabilities $P(y | x)$.
- Optimization methods:
 - Iterative scaling
 - Gradient ascent
 - **Conjugate gradient ascent**
- Maximum-entropy properties of log-linear models
- Smoothing methods using Gaussian prior, and feature selection methods

References

- [Altun, Tsochantaridis, and Hofmann, 2003] Altun, Y., I. Tsochantaridis, and T. Hofmann. 2003. Hidden Markov Support Vector Machines. In *Proceedings of ICML 2003*.
- [Bartlett 1998] P. L. Bartlett. 1998. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2): 525-536, 1998.
- [Bod 98] Bod, R. (1998). *Beyond Grammar: An Experience-Based Theory of Language*. CSLI Publications/Cambridge University Press.
- [Booth and Thompson 73] Booth, T., and Thompson, R. 1973. Applying probability measures to abstract languages. *IEEE Transactions on Computers*, C-22(5), pages 442-450.
- [Borthwick et al 98] Borthwick, A., Sterling, J., Agichtein, E., and Grishman, R. (1998). Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition. *Proc. of the Sixth Workshop on Very Large Corpora*.
- [Collins and Duffy 2001] Collins, M. and Duffy, N. (2001). Convolution Kernels for Natural Language. In *Proceedings of NIPS 14*.
- [Collins and Duffy 2002] Collins, M. and Duffy, N. (2002). New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron. In *Proceedings of ACL 2002*.
- [Collins 2002a] Collins, M. (2002a). Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with the Perceptron Algorithm. In *Proceedings of EMNLP 2002*.
- [Collins 2002b] Collins, M. (2002b). Parameter Estimation for Statistical Parsing Models: Theory and Practice of Distribution-Free Methods. To appear as a book chapter.
- [Crammer and Singer 2001a] Crammer, K., and Singer, Y. 2001a. On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. In *Journal of Machine Learning Research*, 2(Dec):265-292.
- [Crammer and Singer 2001b] Koby Crammer and Yoram Singer. 2001b. Ultraconservative Online Algorithms for Multiclass Problems In *Proceedings of COLT 2001*.
- [Freund and Schapire 99] Freund, Y. and Schapire, R. (1999). Large Margin Classification using the Perceptron Algorithm. In *Machine Learning*, 37(3):277-296.
- [Helmbold and Warmuth 95] Helmbold, D., and Warmuth, M. On Weak Learning. *Journal of Computer and System Sciences*, 50(3):551-573, June 1995.

Experiment	Accuracy	Training Time	# of Features
ME Default	95.5%	15 min	2350
ME IFS	95.8%	15 hours	356
DT Default	91.6%	18 hours	
DT Tuned	92.1%	10 hours	

Table 8.4: Text Categorization Performance on the acq category

- [Hopcroft and Ullman 1979] Hopcroft, J. E., and Ullman, J. D. 1979. *Introduction to automata theory, languages, and computation*. Reading, Mass.: Addison-Wesley.
- [Johnson et al. 1999] Johnson, M., Geman, S., Canon, S., Chi, S., & Riezler, S. (1999). Estimators for stochastic "unification-based" grammars. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. San Francisco: Morgan Kaufmann.
- [Lafferty et al. 2001] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML-01*, pages 282-289, 2001.
- [Littlestone and Warmuth, 1986] Littlestone, N., and Warmuth, M. 1986. Relating data compression and learnability. *Technical report, University of California, Santa Cruz*.
- [MSM93] Marcus, M., Santorini, B., & Marcinkiewicz, M. (1993). Building a large annotated corpus of english: The Penn treebank. *Computational Linguistics*, 19, 313-330.
- [McCallum et al. 2000] McCallum, A., Freitag, D., and Pereira, F. (2000) Maximum entropy markov models for information extraction and segmentation. In *Proceedings of ICML 2000*.
- [Miller et al. 2000] Miller, S., Fox, H., Ramshaw, L., and Weischedel, R. 2000. A Novel Use of Statistical Parsing to Extract Information from Text. In *Proceedings of ANLP 2000*.
- [Ramshaw and Marcus 95] Ramshaw, L., and Marcus, M. P. (1995). Text Chunking Using Transformation-Based Learning. In *Proceedings of the Third ACL Workshop on Very Large Corpora*, Association for Computational Linguistics, 1995.
- [Ratnaparkhi 96] A maximum entropy part-of-speech tagger. In *Proceedings of the empirical methods in natural language processing conference*.
- [Schapire et al., 1998] Schapire R., Freund Y., Bartlett P. and Lee W. S. 1998. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651-1686.
- [Zhang, 2002] Zhang, T. 2002. Covering Number Bounds of Certain Regularized Linear Function Classes. In *Journal of Machine Learning Research*, 2(Mar):527-550, 2002.