Lecture 1: COMS E6998-3, Spring 2011

Log-Linear Models

Michael Collins

The Language Modeling Problem

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

  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

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 .

- There are many possible tags in the position ??
  {NN, NNS, Vt, Vi, IN, DT, . . . }
- The task: model the distribution

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

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

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

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

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

- Many "features" of $t_1, \ldots, t_{i-1}, w_1 \ldots w_n$ may be relevant

$$\begin{array}{lcl} P(t_i = \text{NN} & | & w_i = \text{base}) \\ P(t_i = \text{NN} & | & t_{i-1} \text{ is } \texttt{JJ}) \\ P(t_i = \text{NN} & | & w_i \text{ ends in ``e''}) \\ P(t_i = \text{NN} & | & w_i \text{ ends in ``se''}) \\ P(t_i = \text{NN} & | & w_{i-1} \text{ is ``important''}) \\ P(t_i = \text{NN} & | & w_{i+1} \text{ is ``from''}) \end{array}$$

Language Modeling

- x is a "history" $w_1, w_2, \ldots w_{i-1}$, 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** $P(y \mid x)$ for any x, y where $x \in \mathcal{X}, y \in \mathcal{Y}$

Feature Vector Representations

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

Language Modeling

- x is a "history" $w_1, w_2, \ldots w_{i-1}$, 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

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

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

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

- Example features:

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

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

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

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

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

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

Defining Features in Practice

- We had the following "trigram" feature:

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

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

For example:

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

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

$\ldots$

The Full Set of Features in [Ratnaparkhi 96]

- Contextual Features, e.g.,

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

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

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

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

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

The Full Set of Features in [Ratnaparkhi 96]

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

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

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

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

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

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

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

Parameter Vectors

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

$$\sum_k w_k \phi_k(x, y)$$

Log-Linear Models

- We have some input domain $\mathcal{X}$, and a finite label set $\mathcal{Y}$. Aim is to provide a conditional probability $P(y \mid x)$ for any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.
- A feature is a function $\phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$
(Often binary features or indicator functions $\phi : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$).
- 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 $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.
- We also have a **parameter vector** $\vec{w} \in \mathbb{R}^m$

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:

$$\sum_k w_k \phi_k(x, model) = 5.6 \qquad \sum_k w_k \phi_k(x, the) = -3.2$$

$$\sum_k w_k \phi_k(x, is) = 1.5 \qquad \sum_k w_k \phi_k(x, of) = 1.3$$

$$\sum_k w_k \phi_k(x, models) = 4.5 \qquad \dots$$

- We define

$$P(y \mid x, \vec{w}) = \frac{\exp\{\Sigma_k w_k \phi_k(x, y)\}}{\Sigma_{y' \in \mathcal{Y}} \exp\{\Sigma_k w_k \phi_k(x, y')\}}$$

More About Log-Linear Models

- Why the name?

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

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

$$\vec{w}_{ML} = \text{argmax}_{\vec{w} \in \mathbb{R}^m} L(\vec{w})$$

where

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

Gradient Ascent Methods

- Need to maximize $L(\vec{w})$ where

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

Initialization: $\vec{w} = 0$

Iterate until convergence:

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

Calculating the Maximum-Likelihood Estimates

- Need to maximize:

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

- Calculating gradients:

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

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

$$\texttt{calc\_gradient}(\vec{w}) \rightarrow \left(L(\vec{w}), \left.\frac{dL}{d\vec{w}}\right|_{\vec{w}} \right)$$

  and that's about it!

Overview

- Log-linear models
- Smoothing, feature selection etc. in log-linear models

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.

Smoothing in Maximum Entropy Models

- Say we have a feature:

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

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

$$\sum_i \phi_{100}(x_i, y_i) = \sum_i \sum_y p(y \mid x_i, \vec{w}) \phi_{100}(x_i, y)$$

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

Gaussian Priors

- Modified loss function

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

- Calculating gradients:

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

- Can run conjugate gradient methods as before
- Adds a penalty for large weights

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.

[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.