Maximum Entropy Markov Models

Instructor: Wei Xu

Many slides from Michael Collins

Where are we going with this?

- MaxEnt (Logistic Regression)
  - classify a single observation into one of a set of discrete classes
  - can incorporate arbitrary/overlapping features

- HMM (Hidden Markov Models)
  - sequence tagging — assign a class to each element in a sequence
  - independent assumption (cannot incorporate arbitrary/overlapping features)

- Maximum Entropy Markov Models:
  - combines HMM and MaxEnt

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 ________

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:**

$$
\begin{aligned}
q(\text{model}|w_1, \dots w_{i-1}) \quad = \quad & \lambda_1 q_{ML}(\text{model}|w_{i-2} = \text{any}, w_{i-1} = \text{statistical}) + \\
& \lambda_2 q_{ML}(\text{model}|w_{i-1} = \text{statistical}) + \\
& \lambda_3 q_{ML}(\text{model})
\end{aligned}
$$

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

Trigram Models

$$
\begin{aligned}
q(\text{model}|w_1, \ldots w_{i-1}) \quad = \quad & \lambda_1 q_{ML}(\text{model}|w_{i-2} = \text{any}, w_{i-1} = \text{statistical}) + \\
& \lambda_2 q_{ML}(\text{model}|w_{i-1} = \text{statistical}) + \\
& \lambda_3 q_{ML}(\text{model})
\end{aligned}
$$

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

$$
\begin{array}{lcl}
q_{ML}(\text{model} & | & w_{i-2} = \text{any}) \\
q_{ML}(\text{model} & | & w_{i-1} \text{ is an adjective}) \\
q_{ML}(\text{model} & | & w_{i-1} \text{ ends in ``ical''}) \\
q_{ML}(\text{model} & | & \mathit{author} = \text{Chomsky}) \\
q_{ML}(\text{model} & | & \text{``model'' does not occur somewhere in } w_1, \ldots w_{i-1}) \\
q_{ML}(\text{model} & | & \text{``grammatical'' occurs somewhere in } w_1, \ldots w_{i-1})
\end{array}
$$

A Naive Approach

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

This quickly becomes very unwieldy...

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

similar to HMM, but different!

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

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

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

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

Overview

- Log-linear models
- Parameter estimation in log-linear models
- Smoothing/regularization in log-linear models

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 $x \in \mathcal{X}$, $y \in \mathcal{Y}$

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

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 $f_k(x, y) \in \mathbb{R}$
  (Often **binary features** or **indicator functions**
  $f_k(x, y) \in \{0, 1\}$).
- Say we have m features f_k for $k = 1 \ldots m$
  $\Rightarrow$ A **feature vector** $f(x, y) \in \mathbb{R}^m$ 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
- Example features:

$$f_1(x,y) = \begin{cases} 1 & \text{if } y = \texttt{model} \\ 0 & \text{otherwise} \end{cases}$$

unigram feature

$$f_2(x,y) = \begin{cases} 1 & \text{if } y = \texttt{model} \text{ and } w_{i-1} = \texttt{statistical} \\ 0 & \text{otherwise} \end{cases}$$

bigram feature

$$f_3(x,y) = \begin{cases} 1 & \text{if } y = \texttt{model},\ w_{i-2} = \texttt{any},\ w_{i-1} = \texttt{statistical} \\ 0 & \text{otherwise} \end{cases}$$

trigram feature

$$f_4(x,y) = \begin{cases} 1 & \text{if } y = \texttt{model},\ w_{i-2} = \text{any} \\ 0 & \text{otherwise} \end{cases}$$

skip bigram feature

$$f_5(x,y) = \begin{cases} 1 & \text{if } y = \texttt{model},\ w_{i-1} \text{ is an adjective} \\ 0 & \text{otherwise} \end{cases}$$

$$f_6(x,y) = \begin{cases} 1 & \text{if } y = \texttt{model},\ w_{i-1} \text{ ends in ``ical''} \\ 0 & \text{otherwise} \end{cases}$$

$$f_7(x,y) = \begin{cases} 1 & \text{if } y = \texttt{model},\ \text{author} = \text{Chomsky} \\ 0 & \text{otherwise} \end{cases}$$

$$f_8(x,y) = \begin{cases} 1 & \text{if } y = \texttt{model},\ \text{``model'' is not in } w_1, \ldots w_{i-1} \\ 0 & \text{otherwise} \end{cases}$$

$$f_9(x,y) = \begin{cases} 1 & \text{if } y = \texttt{model},\ \text{``grammatical'' is in } w_1, \ldots w_{i-1} \\ 0 & \text{otherwise} \end{cases}$$

Defining Features in Practice

- We had the following "trigram" feature:

$$f_3(x, y) = \begin{cases} 1 & \text{if } y = \texttt{model},\ w_{i-2} = \texttt{any},\ w_{i-1} = \texttt{statistical} \\ 0 & \text{otherwise} \end{cases}$$

- 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

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

Do not include trigrams that are not seen in the training data

index of unique trigrams in training data

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, ...
- We have m features $f_k(x, y)$ for $k = 1 \ldots m$

For example:

$$f_1(x, y) = \begin{cases} 1 & \text{if current word } w_i \text{ is base and } y = \texttt{Vt} \\ 0 & \text{otherwise} \end{cases}$$

word/tag pair

$$f_2(x, y) = \begin{cases} 1 & \text{if current word } w_i \text{ ends in } \texttt{ing} \text{ and } y = \texttt{VBG} \\ 0 & \text{otherwise} \end{cases}$$

$$\ldots$$

The Full Set of Features in Ratnaparkhi, 1996

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

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

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

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

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

The Full Set of Features in Ratnaparkhi, 1996

- Contextual Features, e.g.,

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

trigram tag feature

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

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

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

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

The 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

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

often sparse with few 1's vs. 0's

Parameter Vectors

- Given features $f_k(x, y)$ for $k = 1 \dots m$, also define a **parameter vector** $v \in \mathbb{R}^m$

all possible m-dimensional real value vectors

- Each (x, y) pair is then mapped to a "score"

$$v \cdot f(x, y) = \sum_k v_k f_k(x, y)$$

Recall Logistic/Softmax Regression!

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 ____________
- Each possible y gets a different score:

$$v \cdot f(x, model) = 5.6 \qquad v \cdot f(x, the) = -3.2$$

$$v \cdot f(x, is) = 1.5 \qquad v \cdot f(x, of) = 1.3$$

$$v \cdot f(x, models) = 4.5 \qquad \ldots$$

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 $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$
(Often binary features or indicator functions
$f_k : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$).
- Say we have m features f_k for $k = 1 \ldots m$
$\Rightarrow$ A feature vector $f(x, y) \in \mathbb{R}^m$ for any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.
- We also have a **parameter vector** $v \in \mathbb{R}^m$
- We define

$$p(y \mid x; v) = \frac{e^{v \cdot f(x,y)}}{\sum_{y' \in \mathcal{Y}} e^{v \cdot f(x,y')}}$$

Softmax!

Why the name?

$$\log p(y \mid x; v) = \underbrace{v \cdot f(x, y)}_{\text{Linear term}} - \underbrace{\log \sum_{y' \in \mathcal{Y}} e^{v \cdot f(x, y')}}_{\text{Normalization term}}$$

Overview

- Log-linear models
- Parameter estimation in log-linear models
- Smoothing/regularization in log-linear models

Maximum-Likelihood Estimation

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

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

where

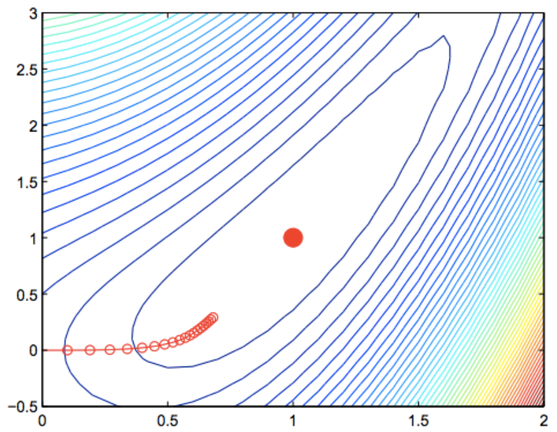
$$L(v) = \sum_{i=1}^{n} \log p(y^{(i)} \mid x^{(i)}; v) = \sum_{i=1}^{n} v \cdot f(x^{(i)}, y^{(i)}) - \sum_{i=1}^{n} \log \sum_{y' \in \mathcal{Y}} e^{v \cdot f(x^{(i)}, y')}$$

concave function!

Calculating the Maximum-Likelihood Estimates

- Need to maximize:

$$L(v) = \sum_{i=1}^{n} v \cdot f(x^{(i)}, y^{(i)}) - \sum_{i=1}^{n} \log \sum_{y' \in \mathcal{Y}} e^{v \cdot f(x^{(i)}, y')}$$



- Calculating gradients:

$$\begin{aligned}
\frac{dL(v)}{dv_k} &= \sum_{i=1}^{n} f_k(x^{(i)}, y^{(i)}) - \sum_{i=1}^{n} \frac{\sum_{y' \in \mathcal{Y}} f_k(x^{(i)}, y') e^{v \cdot f(x^{(i)}, y')}}{\sum_{z' \in \mathcal{Y}} e^{v \cdot f(x^{(i)}, z')}} \\
&= \sum_{i=1}^{n} f_k(x^{(i)}, y^{(i)}) - \sum_{i=1}^{n} \sum_{y' \in \mathcal{Y}} f_k(x^{(i)}, y') \frac{e^{v \cdot f(x^{(i)}, y')}}{\sum_{z' \in \mathcal{Y}} e^{v \cdot f(x^{(i)}, z')}} \\
&= \underbrace{\sum_{i=1}^{n} f_k(x^{(i)}, y^{(i)})}_{\text{Empirical counts}} - \underbrace{\sum_{i=1}^{n} \sum_{y' \in \mathcal{Y}} f_k(x^{(i)}, y') p(y' \mid x^{(i)}; v)}_{\text{Expected counts}}
\end{aligned}$$

Multinominal Logistic Regression

Gradient Ascent Methods

- Need to maximize $L(v)$ where

$$\frac{dL(v)}{dv} = \sum_{i=1}^{n} f(x^{(i)}, y^{(i)}) - \sum_{i=1}^{n} \sum_{y' \in \mathcal{Y}} f(x^{(i)}, y') p(y' \mid x^{(i)}; v)$$

Initialization: $v = 0$

Iterate until convergence:

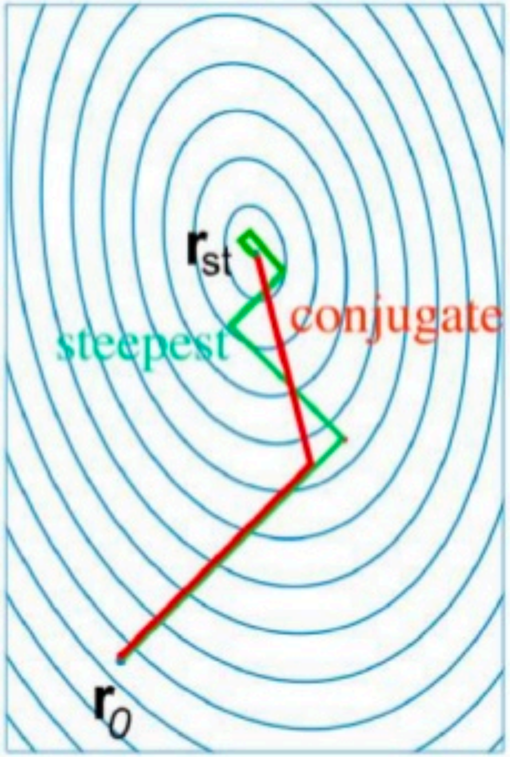
- Calculate $\Delta = \frac{dL(v)}{dv}$
- Calculate $\beta_* = \text{argmax}_\beta L(v + \beta\Delta)$ (Line Search)
- Set $v \leftarrow v + \beta_*\Delta$

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

$$\texttt{calc\_gradient}(v) \rightarrow \left(L(v), \frac{dL(v)}{dv} \right)$$

and that's about it!



e.g. LBFGS Algorithm (Limited-memory Broyden–Fletcher–Goldfarb–Shanno)

https://www.youtube.com/watch?v=h4cG8jLGmKg