Maximum Entropy Models

Pawan Goyal

CSE, IIT Kharagpur

Week 4, Lecture 3

Issues with Markov Model Tagging

Unknown Words

We do not have the required probabilities.

Issues with Markov Model Tagging

Unknown Words

We do not have the required probabilities.

Possible solutions:

Issues with Markov Model Tagging

Unknown Words

We do not have the required probabilities.

Possible solutions:

- Use morphological cues (capitalization, suffix) to assign a more calculated guess

Issues with Markov Model Tagging

Unknown Words

We do not have the required probabilities.

Possible solutions:

- Use morphological cues (capitalization, suffix) to assign a more calculated guess

Limited Context

- "is clearly **marked**" → verb, past participle
- "he clearly **marked**" → verb, past tense

Issues with Markov Model Tagging

Unknown Words

We do not have the required probabilities.

Possible solutions:

- Use morphological cues (capitalization, suffix) to assign a more calculated guess

Limited Context

- “is clearly **marked**” → verb, past participle
- “he clearly **marked**” → verb, past tense

Possible solution:

Issues with Markov Model Tagging

Unknown Words

We do not have the required probabilities.

Possible solutions:

- Use morphological cues (capitalization, suffix) to assign a more calculated guess

Limited Context

- "is clearly **marked**" → verb, past participle
- "he clearly **marked**" → verb, past tense

Possible solution: Use higher order model, combine various n-gram models to avoid sparseness problem

Maximum Entropy Modeling: Discriminative Model

Maximum Entropy Modeling: Discriminative Model

- We may identify a heterogeneous set of features which contribute in some way to the choice of POS tag of the current word.

Maximum Entropy Modeling: Discriminative Model

- We may identify a heterogeneous set of features which contribute in some way to the choice of POS tag of the current word.
  - Whether it is the first word in the article

Maximum Entropy Modeling: Discriminative Model

- We may identify a heterogeneous set of features which contribute in some way to the choice of POS tag of the current word.
  - Whether it is the first word in the article
  - Whether the next word is *to*

Maximum Entropy Modeling: Discriminative Model

- We may identify a heterogeneous set of features which contribute in some way to the choice of POS tag of the current word.
  - Whether it is the first word in the article
  - Whether the next word is *to*
  - Whether one of the last 5 words is a preposition, etc.
- MaxEnt combines these features in a probabilistic model

Maximum Entropy: The Model

$$p_\lambda(y|x) = \frac{1}{Z_\lambda(x)} exp\left(\sum_i \lambda_i f_i(x,y)\right)$$

Maximum Entropy: The Model

$$p_\lambda(y|x) = \frac{1}{Z_\lambda(x)} exp\left(\sum_i \lambda_i f_i(x,y)\right)$$

where

- $Z_\lambda(x)$ is a normalizing constant given by

$$Z_\lambda(x) = \sum_y exp\left(\sum_i \lambda_i f_i(x,y)\right)$$

Maximum Entropy: The Model

$$p_\lambda(y|x) = \frac{1}{Z_\lambda(x)} exp\left(\sum_i \lambda_i f_i(x,y)\right)$$

where

- $Z_\lambda(x)$ is a normalizing constant given by

$$Z_\lambda(x) = \sum_y exp\left(\sum_i \lambda_i f_i(x,y)\right)$$

- λ_i is a weight given to a feature f_i

Maximum Entropy: The Model

$$p_\lambda(y|x) = \frac{1}{Z_\lambda(x)} exp\left(\sum_i \lambda_i f_i(x,y)\right)$$

where

- $Z_\lambda(x)$ is a normalizing constant given by

$$Z_\lambda(x) = \sum_y exp\left(\sum_i \lambda_i f_i(x,y)\right)$$

- λ_i is a weight given to a feature f_i
- x denotes an observed datum and y denotes a class

What is the form of the features?

Features in Maximum Entropy Models

- Features encode elements of the context x for predicting tag y
- Context x is taken around the word w, for which a tag y is to be predicted

Features in Maximum Entropy Models

- Features encode elements of the context x for predicting tag y
- Context x is taken around the word w, for which a tag y is to be predicted
- Features are binary values functions, e.g.,

$$f(x,y) = \left\{ \begin{array}{ll} 1 & \text{if } isCapitalized(w) \& y = NNP \\ 0 & otherwise \end{array} \right\}$$

Example Features

Example: Named Entities

- LOCATION (in Arcadia)
- LOCATION (in Québec)
- DRUG (taking Zantac)
- PERSON (saw Sue)

Example Features

Example: Named Entities

- LOCATION (in Arcadia)
- LOCATION (in Québec)
- DRUG (taking Zantac)
- PERSON (saw Sue)

Example Features

- $f_1(x,y) = [y = LOCATION \wedge w_{-1} = \text{“}in\text{”} \wedge isCapitalized(w)]$
- $f_2(x,y) = [y = LOCATION \wedge hasAccentedLatinChar(w)]$
- $f_3(x,y) = [y = DRUG \wedge ends(w, \text{“}c\text{”})]$

Tagging with Maximum Entropy Model

- $W = w_1 \ldots w_n$ - words in the corpus (observed)
- $T = t_1 \ldots t_n$ - the corresponding tags (unknown)

Tagging with Maximum Entropy Model

- $W = w_1 \dots w_n$ - words in the corpus (observed)
- $T = t_1 \dots t_n$ - the corresponding tags (unknown)

Tag sequence candidate $\{t_1, \dots, t_n\}$ has conditional probability:

$$P(t_1, \dots, t_n | w_1 \dots, w_n) = \prod_{i=1}^{n} p(t_i | x_i)$$

Tagging with Maximum Entropy Model

- $W = w_1 \ldots w_n$ - words in the corpus (observed)
- $T = t_1 \ldots t_n$ - the corresponding tags (unknown)

Tag sequence candidate $\{t_1, \ldots, t_n\}$ has conditional probability:

$$P(t_1, \ldots, t_n | w_1 \ldots, w_n) = \prod_{i=1}^{n} p(t_i | x_i)$$

- The context x_i also includes previously assigned tags for a fixed history.
- Beam search is used to find the most probable sequence

Beam Inference

Beam Inference

- At each position, keep the top k complete sequences
- Extend each sequence in each local way
- The extensions compete for the k slots at the next position

Beam Inference

Beam Inference

- At each position, keep the top k complete sequences
- Extend each sequence in each local way
- The extensions compete for the k slots at the next position

But what is a MaxEnt model?

Let's go to the basics now!

Maximum Entropy Model

Intuitive Principle

Model all that is known and assume nothing about that which is unknown.

Maximum Entropy Model

Intuitive Principle

Model all that is known and assume nothing about that which is unknown.
Given a collection of facts, choose a model which is consistent with all the facts, but otherwise as uniform as possible.

Maximum Entropy: Overview

- Suppose we wish to model an expert translator's decisions concerning the proper French rendering of the English word '*in*'.

Maximum Entropy: Overview

- Suppose we wish to model an expert translator's decisions concerning the proper French rendering of the English word '*in*'.
- Each French word or phrase f is assigned an estimate $p(f)$, probability that the expert would choose f as a translation of '*in*'.

Maximum Entropy: Overview

- Suppose we wish to model an expert translator's decisions concerning the proper French rendering of the English word '*in*'.
- Each French word or phrase f is assigned an estimate $p(f)$, probability that the expert would choose f as a translation of '*in*'.
- Collect a large sample of instances of the expert's decisions

Maximum Entropy: Overview

- Suppose we wish to model an expert translator's decisions concerning the proper French rendering of the English word '*in*'.
- Each French word or phrase f is assigned an estimate $p(f)$, probability that the expert would choose f as a translation of '*in*'.
- Collect a large sample of instances of the expert's decisions
- **Goal**: extract a set of facts about the decision-making process (first task) that will aid in constructing a model of this process (second task)

Maximum Entropy Model: Overview

First clue: list of allowed translations

- Suppose the translator always chooses among {dans, en, á, au cours de, pendant}.

Maximum Entropy Model: Overview

First clue: list of allowed translations

- Suppose the translator always chooses among {dans, en, á, au cours de, pendant}.
- First constraint: p(dans)+p(en)+p(á)+p(au cours de)+p(pendant) = 1.

Maximum Entropy Model: Overview

First clue: list of allowed translations

- Suppose the translator always chooses among {dans, en, á, au cours de, pendant}.
- First constraint: p(dans)+p(en)+p(á)+p(au cours de)+p(pendant) = 1.
- Infinite number of models p for which this identity holds, the most intuitive model?

Maximum Entropy Model: Overview

First clue: list of allowed translations

- Suppose the translator always chooses among {dans, en, á, au cours de, pendant}.
- First constraint: p(dans)+p(en)+p(á)+p(au cours de)+p(pendant) = 1.
- Infinite number of models p for which this identity holds, the most intuitive model?
- *allocate the total probability evenly among the five possible phrases* $\rightarrow$ most uniform model subject to our knowledge.

Maximum Entropy Model: Overview

First clue: list of allowed translations

- Suppose the translator always chooses among {dans, en, á, au cours de, pendant}.
- First constraint: p(dans)+p(en)+p(á)+p(au cours de)+p(pendant) = 1.
- Infinite number of models p for which this identity holds, the most intuitive model?
- *allocate the total probability evenly among the five possible phrases* → most uniform model subject to our knowledge.
- *Is it the most uniform model overall?*

Maximum Entropy Model: Overview

First clue: list of allowed translations

- Suppose the translator always chooses among {dans, en, á, au cours de, pendant}.
- First constraint: p(dans)+p(en)+p(á)+p(au cours de)+p(pendant) = 1.
- Infinite number of models p for which this identity holds, the most intuitive model?
- *allocate the total probability evenly among the five possible phrases* → most uniform model subject to our knowledge.
- *Is it the most uniform model overall?* → No, that would grant an equal probability to every *possible* French phrase.

Maximum Entropy Model: Overview

More clues from the expert's decision

- **Second clue:** Suppose the expert chose either '*dans*' or '*en*' 30% of the time.

Maximum Entropy Model: Overview

More clues from the expert's decision

- **Second clue:** Suppose the expert chose either '*dans*' or '*en*' 30% of the time.
- **Third clue:** In half of the cases, the expert chose either '*dans*' or '*á*'

Maximum Entropy Model: Overview

More clues from the expert's decision

- **Second clue:** Suppose the expert chose either '*dans*' or '*en*' 30% of the time.
- **Third clue:** In half of the cases, the expert chose either '*dans*' or '*á*'

How do we measure uniformity of a model?

As we add complexity to the model, we face two difficulties:

- What exactly is meant by "uniform"?
- How can one measure the uniformity of a model?

Maximum Entropy Modeling

Entropy: measures the uncertainty of a distribution.

Quantifying uncertainty ("surprise")

- Event x
- Probability p_x
- Surprise: $log(1/p_x)$

Maximum Entropy Modeling

Entropy: measures the uncertainty of a distribution.

Quantifying uncertainty ("surprise")

- Event x
- Probability p_x
- Surprise: $log(1/p_x)$

Entropy: expected surprise (over p)

$$H(p) = E_p\left[log_2 \frac{1}{p_x}\right] = -\sum_x p_x log_2 p_x$$

Maximum Entropy Modeling

Entropy: measures the uncertainty of a distribution.

Quantifying uncertainty ("surprise")

- Event x
- Probability p_x
- Surprise: $log(1/p_x)$

Entropy: expected surprise (over p)

$$H(p) = E_p\left[log_2\frac{1}{p_x}\right] = -\sum_x p_x log_2 p_x$$

Coin Tossing

Maximum Entropy Modeling

Distribution required

- Minimize commitment = maximize entropy
- Resemble some reference distribution

Maximum Entropy Modeling

Distribution required

- Minimize commitment = maximize entropy
- Resemble some reference distribution

Solution

Maximize entropy H, subject to feature-based constraints:

$$E_p[f_i] = E_{\tilde{p}}[f_i]$$

Maximum Entropy Modeling

Distribution required

- Minimize commitment = maximize entropy
- Resemble some reference distribution

Solution

Maximize entropy H, subject to feature-based constraints:

$$E_p[f_i] = E_{\tilde{p}}[f_i]$$

Adding constraints

- Lowers maximum entropy
- Brings the distribution further from uniform and closer to data

Maximum Entropy Principle

Given n feature functions f_i, we would like p to lie in the subset C of P defined by

$$C = \{p \in P | p(f_i) = \tilde{p}(f_i), i \in \{1, 2, \ldots, n\}\}$$

Maximum Entropy Principle

Given n feature functions f_i, we would like p to lie in the subset C of P defined by

$$C = \{p \in P | p(f_i) = \tilde{p}(f_i), i \in \{1, 2, \ldots, n\}\}$$

Empirical count (expectation) of a feature

$$\tilde{p}(f_i) = \sum_{x,y} \tilde{p}(x,y) f_i(x,y)$$

Maximum Entropy Principle

Given n feature functions f_i, we would like p to lie in the subset C of P defined by

$$C = \{p \in P | p(f_i) = \tilde{p}(f_i), i \in \{1, 2, \ldots, n\}\}$$

Empirical count (expectation) of a feature

$$\tilde{p}(f_i) = \sum_{x,y} \tilde{p}(x,y) f_i(x,y)$$

Model expectation of a feature

$$p(f_i) = \sum_{x,y} \tilde{p}(x) p(y|x) f_i(x,y)$$

Select the distribution which is most uniform (conditional probability):

$$p^* = argmax_{p \in C} H(p) = H(Y|X) \approx -\sum_{x,y} \tilde{p}(x) p(y|x) log p(y|x)$$

Maximum Entropy Principle

$$p^* = argmax_{p \in C} H(p)$$

Maximum Entropy Principle

$$p^* = argmax_{p \in C} H(p)$$

Constraint Optimization

Introduce a parameter λ_i for each feature f_i. Lagrangian is given by

$$\wedge(p,\lambda) = H(p) + \sum_i \lambda_i (p(f_i) - \tilde{p}(f_i))$$

Solving, we get

$$p_\lambda(y|x) = \frac{1}{Z_\lambda(x)} exp\left(\sum_i \lambda_i f_i(x,y)\right)$$

where $Z_\lambda(x)$ is a normalizing constant given by

$$Z_\lambda(x) = \sum_y exp\left(\sum_i \lambda_i f_i(x,y)\right)$$