Maximum Entropy Model (I)

LING 572

Fei Xia

Week 4: 01/28-02/02/2010

MaxEnt in NLP

- The maximum entropy principle has a long history.
- The MaxEnt algorithm was introduced to the NLP field by Berger et. al. (1996).
- Used in many NLP tasks: Tagging, Parsing, PP attachment, …

Reference papers

- (Ratnaparkhi, 1997)
- (Berger et. al., 1996)
- (Ratnaparkhi, 1996)
- (Klein and Manning, 2003)

People often choose different notations.

Notation

	Input	Output	The pair
(Berger et. al., 1996)	x	y	(x,y)
(Ratnaparkhi, 1997)	b	a	x
(Ratnaparkhi, 1996)	h	t	(h,t)
(Klein and Manning, 2003)	d	c	(c,d)

We following the notation in (Berger et al., 1996)

Outline

- Overview
- The Maximum Entropy Principle

- Modeling**
- Decoding

- Training**

- Case study: POS tagging

The Overview

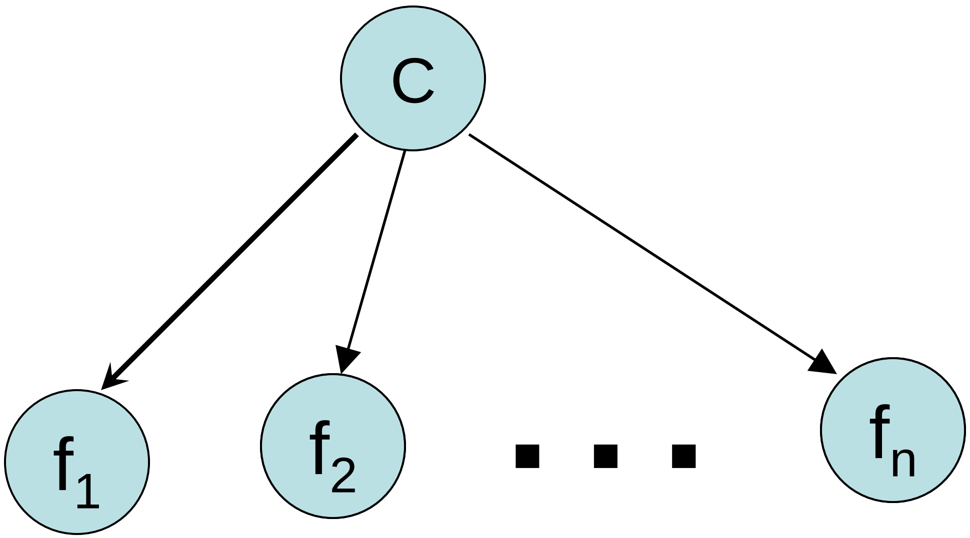
Joint vs. Conditional models **

- Given training data {(x,y)}, we want to build a model to predict y for new x's. For each model, we need to estimate the parameters θ.

- Joint (generative) models estimate P(x,y) by maximizing the likelihood: $P(X,Y|\theta)$
  - Ex: n-gram models, HMM, Naïve Bayes, PCFG
  - Choosing weights is trivial: just use relative frequencies.

- Conditional (discriminative) models estimate P(y|x) by maximizing the conditional likelihood: $P(Y|X, \theta)$
  - Ex: MaxEnt, SVM, etc.
  - Choosing weights is harder.

Naïve Bayes Model



Assumption: each f_m is conditionally independent from f_n given C.

The conditional independence assumption

f_m and f_n are conditionally independent given c:

$$P(f_m \mid c, f_n) = P(f_m \mid c)$$

Counter-examples in the text classification task:

- P("bank" | politics) != P("bank" | politics, "bailout")

Q: How to deal with correlated features?

A: Many models, including MaxEnt, do not assume that features are conditionally independent.

Naïve Bayes highlights

- Choose
  $c^* = \arg\max_c P(c) \prod_k P(f_k \mid c)$

- Two types of model parameters:
  - Class prior: $P(c)$
  - Conditional probability: $P(f_k \mid c)$

- The number of model parameters:
  $|C|+|CV|$

$P(f \mid c)$ in NB

	f_1	f_2	…	f_j
c_1	$P(f_1 \mid c_1)$	$P(f_2 \mid c_1)$	…	$P(f_j \mid c_1)$
c_2	$P(f_1 \mid c_2)$	…	…	…
…	…			
c_i	$P(f_1 \mid c_i)$	…	…	$P(f_j \mid c_i)$

Each cell is a weight for a particular (class, feat) pair.

Weights in NB and MaxEnt

- In NB
  - P(f | y) are probabilities (i.e., $\in$ [0,1])
  - P(f | y) are multiplied at test time

$$P(y|x) = \frac{P(y)\prod_k P(f_k|y)}{Z}$$

$$= \frac{e^{ln P(y) + \sum_k ln P(f_k|y)}}{Z}$$

- In MaxEnt
  - the weights are real numbers: they can be negative.
  - the weights are added at test time

$$P(y|x) = \frac{e^{\sum_j \lambda_j f_j(x,y)}}{Z}$$

The highlights in MaxEnt

$$P(y|x) = \frac{e^{\sum_j \lambda_j \ f_j(x,y)}}{Z}$$

$f_j(x,y)$ is a feature function, which normally corresponds to a (feature, class) pair.

Training: to estimate λ_j

Testing: to calculate P(y|x)

Main questions

- What is the maximum entropy principle?
- What is a feature function?
- Modeling: Why does P(y|x) have the form?

$$P(y|x) = \frac{e^{\sum_j \lambda_j \ f_j(x,y)}}{Z}$$

- Training: How do we estimate λ_j ?

Outline

- Overview
- The Maximum Entropy Principle

- Modeling**
- Decoding

- Training*

- Case study

The maximal entropy principle

The maximum entropy principle

- Related to Occam's razor and other similar justifications for scientific inquiry

- Make the minimum assumptions about unseen data.

- Also: Laplace's *Principle of Insufficient Reason: when one has no* information to distinguish between the probability of two events, the best strategy is to consider them equally likely.

Maximum Entropy

- Why maximum entropy?
  - Maximize entropy = Minimize commitment

- Model all that is known and assume nothing about what is unknown.
  - Model all that is known: satisfy a set of constraints that must hold

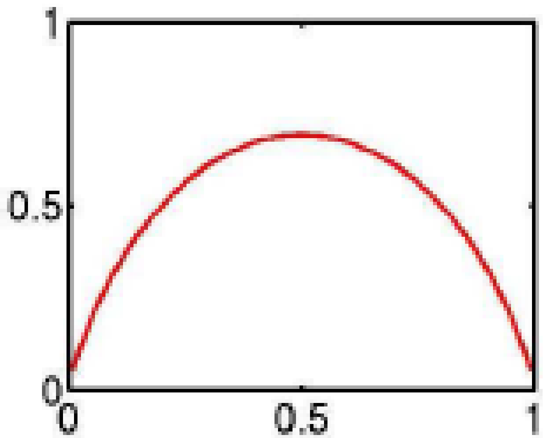
  - Assume nothing about what is unknown:
    choose the most “uniform” distribution
    ➔ choose the one with maximum entropy

Ex1: Coin-flip example (Klein & Manning, 2003)

- Toss a coin: p(H)=p1, p(T)=p2.
- Constraint: p1 + p2 = 1
- Question: what's p(x)? That is, what is the value of p1?
- Answer: choose the p that maximizes H(p)

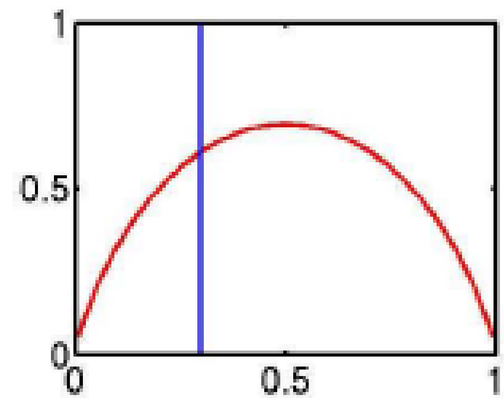
$$H(p) = -\sum_{x} p(x) \log p(x)$$

H



p1

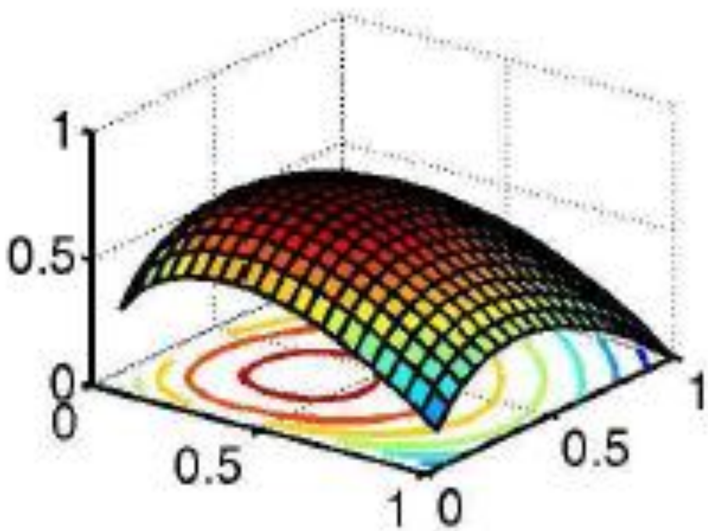
H



p1=0.3

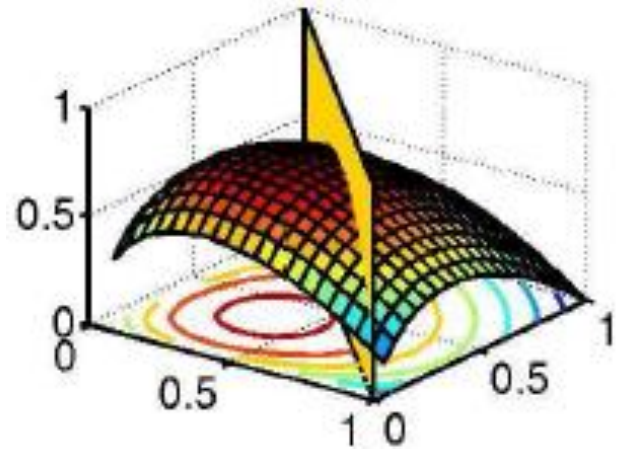
Coin-flip example** (cont)

H

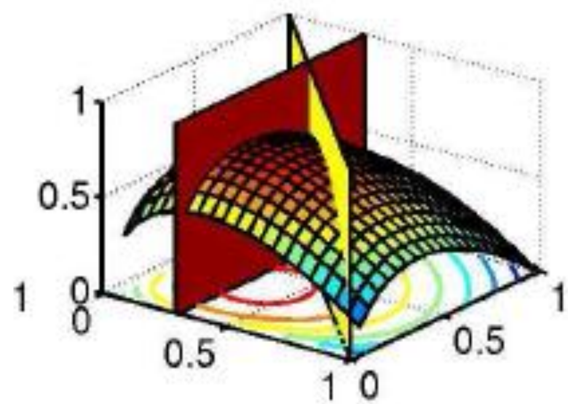


p1

p2



p1 + p2 = 1



p1+p2=1.0, p1=0.3

Ex2: An MT example (Berger et. al., 1996)

Possible translation for the word “in” is:

$$\{dans,\ en,\ \grave{a},\ au\ cours\ de,\ pendant\}$$

Constraint:

$$p(dans) + p(en) + p(\grave{a}) + p(au\ cours\ de) + p(pendant) = 1$$

Intuitive answer:

$$\begin{aligned} p(dans) &= 1/5 \\ p(en) &= 1/5 \\ p(\grave{a}) &= 1/5 \\ p(au\ cours\ de) &= 1/5 \\ p(pendant) &= 1/5 \end{aligned}$$

An MT example (cont)

Constraints:

$$
\begin{aligned}
p(dans) + p(en) &= 3/10 \\
p(dans) + p(en) + p(\grave{a}) + p(au\ cours\ de) + p(pendant) &= 1
\end{aligned}
$$

Intuitive answer:

$$
\begin{aligned}
p(dans) &= 3/20 \\
p(en) &= 3/20 \\
p(\grave{a}) &= 7/30 \\
p(au\ cours\ de) &= 7/30 \\
p(pendant) &= 7/30
\end{aligned}
$$

An MT example (cont)

Constraints:

$$\begin{aligned} p(dans) + p(en) &= 3/10 \\ p(dans) + p(en) + p(à) + p(au\ cours\ de) + p(pendant) &= 1 \\ p(dans) + p(à) &= 1/2 \end{aligned}$$

Intuitive answer:

??

Ex3: POS tagging (Klein and Manning, 2003)

- Lets say we have the following event space:

NN	NNS	NNP	NNPS	VBZ	VBD

- … and the following empirical data:

3	5	11	13	3	1

- Maximize H:

$1/e$	$1/e$	$1/e$	$1/e$	$1/e$	$1/e$

- … want probabilities: E[NN,NNS,NNP,NNPS,VBZ,VBD] = 1

1/6	1/6	1/6	1/6	1/6	1/6

Ex3 (cont)

- Too uniform!

- N* are more common than V*, so we add the feature f_N = {NN, NNS, NNP, NNPS}, with E[f_N] =32/36

NN	NNS	NNP	NNPS	VBZ	VBD
8/36	8/36	8/36	8/36	2/36	2/36

- … and proper nouns are more frequent than common nouns, so we add f_P = {NNP, NNPS}, with E[f_P] =24/36

NN	NNS	NNP	NNPS	VBZ	VBD
4/36	4/36	12/36	12/36	2/36	2/36

Ex4: Overlapping features (Klein and Manning, 2003)

Empirical

	A	a
B	1	1
b	1	0

	A	a
B		
b		

All = 1

	A	a
B	p1	p2
b	p3	p4

	A	a
B	1/4	1/4
b	1/4	1/4

Ex4 (cont)

Empirical

	A	a
B	1	1
b	1	0

	A	a
B	p1	p2
b	$\frac{2}{3} - p_1$	$\frac{1}{3} - p_2$

	A	a
B		
b		

A = 2/3

	A	a
B	1/3	1/6
b	1/3	1/6

Ex4 (cont)

Empirical

	A	a
B	1	1
b	1	0

	A	a
B		
b		

A = 2/3

	A	a
B		
b		

B = 2/3

	A	a
B	p1	$\frac{2}{3} - p_1$
b	$\frac{2}{3} - p_1$	$p_1 - \frac{1}{3}$

	A	a
B	4/9	2/9
b	2/9	1/9

The MaxEnt Principle summary

- Goal: Among all the distributions that satisfy the constraints, choose the one, p*, that maximizes H(p).

$$p^* = \arg\max_{p \in P} H(p)$$

- Q1: How to represent constraints?

- Q2: How to find such distributions?

Reading #2

(Q1): Let $P(X=i)$ be the probability of getting an i when rolling a dice. What is the value of $P(x)$ with the maximum entropy if the following is true?

(a) $P(X=1) + P(X=2) = ½$

¼, ¼, 1/8, 1/8, 1/8, 1/8

(b) $P(X=1) + P(X=2) = 1/2$ and $P(X=6) = 1/3$

¼, ¼, 1/18, 1/18, 1/18, 1/3

(Q2) In the text classification task, |V| is the number of features, |C| is the number of classes. How many feature functions are there?

|C| * |V|

Outline

- Overview
- The Maximum Entropy Principle
- Modeling**
- Decoding
- Training**
- Case study

Modeling

The Setting

- From the training data, collect (x, y) pairs:
  - $x \in X$: the observed data
  - $y \in Y$: thing to be predicted (e.g., a class in a classification problem)
  - Ex: In a text classification task
    - x: a document
    - y: the category of the document

- To estimate $P(y|x)$

The basic idea

- Goal: to estimate p(y|x)

- Choose p(x,y) with maximum entropy (or "uncertainty") subject to the constraints (or "evidence").

$$H(p) = -\sum_{(x,y)\in X\times Y} p(x,y)\log p(x,y)$$

The outline for modeling

- Feature function: $f_j(x, y)$
- Calculating the expectation of a feature function
- The forms of P(x,y) and P(y|x)

Feature function

The definition

- A feature function is a binary-valued function on events:

$$f_j : X \times Y \rightarrow \{0,1\}$$

The j in f_j corresponds to a (feature, class) pair (t, c)

$f_j(x, y) = 1$ iff t is present in x and $y = c$.

- Ex:

$$f_j(x, y) = \begin{cases} 1 & if \;\; y = Politics \;\&\; x \; contains \;\; "the" \\ 0 & o.w. \end{cases}$$

The weights in NB

	f_1	f_2	…	f_k
c_1				
c_2				
…				
c_i				

The weights in NB

	f_1	f_2	…	f_j
c_1	$P(f_1 \mid c_1)$	$P(f_2 \mid c_1)$	…	$P(f_j \mid c_1)$
c_2	$P(f_1 \mid c_2)$	…	…	…
…	…			
c_i	$P(f_1 \mid c_i)$	…	…	$P(f_j \mid c_i)$

Each cell is a weight for a particular (class, feat) pair.

The matrix in MaxEnt

	$t_{\mathbf{1}}$	$t_{\mathbf{2}}$	**…**	$t_{\mathbf{k}}$
c_1	f_1	f_2	…	f_k
c_2	f_{k+1}	f_{k+2}	…	f_{2k}
…	…			
c_i	$f_{k*(i-1)+1}$			f_{k*i}

Each feature function f_j corresponds to a (feat, class) pair.

The weights in MaxEnt

	t_1	t_2	…	t_k
c_1	λ_1	λ_2	…	λ_k
c_2	…	…	…	…
…	…			
c_i	…			λ_{ki}

Each feature function f_j has a weight λ_j.

Feature function summary

- A feature function in MaxEnt corresponds to a (feat, class) pair.
- The number of feature functions in MaxEnt is approximately |C| * |V|.
- A MaxEnt trainer is to learn the weights for the feature functions.

The outline for modeling

- Feature function: $f_j(x, y)$

- Calculating the expectation of a feature function

- The forms of P(x,y) and P(y | x)

The expected return

- Ex1:
  - Flip a coin
    - if it is a head, you will get 100 dollars
    - if it is a tail, you will lose 50 dollars
  - What is the expected return?

  P(X=H) * 100 + P(X=T) * (-50)

- Ex2:
  - If it is a x_i, you will receive v_i dollars?
  - What is the expected return?

$$\sum_i P(X = x_i)\ v_i$$

Calculating the expectation of a function

Let $P(X = x)$ be a distribution of a random variable X.
Let $f(x)$ be a function of x.
Let $E_p(f)$ be the expectation of $f(x)$ based on $P(x)$.

$$E_P(f) = \sum_i P(X = x_i)\boxed{f(x_i)}$$

$$\sum_i P(X = x_i)\boxed{v_i}$$

Empirical expectation

- Denoted as : $\tilde{p}(x)$

- Ex1: Toss a coin four times and get H, T, H, and H.

- The average return: (100-50+100+100)/4 = $\boxed{62.5}$

- Empirical distribution:

$$\tilde{p}(X = H) = 3/4$$
$$\tilde{p}(X = T) = 1/4$$

- Empirical expectation:

$\frac{3}{4} * 100 + \frac{1}{4} * (-50) = \boxed{62.5}$

Model expectation

- Ex1: Toss a coin four times and get H, T, H, and H.

- A model: $p(x)$

$$p(X = H) = 1/2$$

$$p(X = T) = 1/2$$

- Model expectation:

  1/2 * 100 + 1/2 * (-50) = 25

Some notations

Training data: S

Empirical distribution: $\tilde{p}(x, y)$

A model: $p(x, y)$

The j^th feature function: $f_j(x, y)$

Empirical expectation of f_j

$$E_{\tilde{p}} f_j = \sum_{(x,y)\in X\times Y} \tilde{p}(x, y) f_j(x, y)$$

Model expectation of f_j

$$E_p f_j = \sum_{(x,y)\in X\times Y} p(x, y) f_j(x, y)$$

Empirical expectation**

$$E_{\tilde{p}} f_j = \sum_{x \in X, y \in Y} \tilde{p}(x, y) f_j(x, y)$$

$$= \sum_{x \in X, y \in Y} \tilde{p}(x) \tilde{p}(y \mid x) f_j(x, y) = \sum_{x \in X} \tilde{p}(x) \sum_{y \in Y} \tilde{p}(y \mid x) f_j(x, y)$$

$$= \sum_{x \in S} \tilde{p}(x) \sum_{y \in Y} \tilde{p}(y \mid x) f_j(x, y) = \frac{1}{N} \sum_{i=1}^{N} \sum_{y \in Y} \tilde{p}(y \mid x_i) f_j(x_i, y)$$

$$= \frac{1}{N} \sum_{i=1}^{N} f_j(x_i, y_i)$$

An example

- Training data:

x1 c1 t1 t2 t3

x2 c2 t1 t4

x3 c1 t3 t4

x4 c3 t1 t3

Raw counts $\sum_{i=1}^{N} f_j(x_i, y_i)$

	t1	t2	t3	t4
c1	1	1	2	1
c2	1	0	0	1
c3	1	0	1	0

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

An example

- Training data:

  x1 c1 t1 t2 t3

  x2 c2 t1 t4

  x3 c1 t3 t4

  x4 c3 t1 t3

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

Empirical expectation

	t1	t2	t3	t4
c1	1/4	1/4	2/4	1/4
c2	1/4	0/4	0/4	1/4
c3	1/4	0/4	1/4	0/4

Calculating empirical expectation

Let N be the number of training instances

for each instance x in the training data

    let y be the true class label of x

    for each feature t in x

        empirical_expect [t] [y] += 1/N

Model expectation**

$$E_p f_j = \sum_{x \in X, y \in Y} p(x,y) f_j(x,y)$$

$$= \sum_{x \in X, y \in Y} p(x) p(y \mid x) f_j(x,y) \qquad \approx \sum_{x \in X, y \in Y} \tilde{p}(x) p(y \mid x) f_j(x,y)$$

$$= \sum_{x \in X} \tilde{p}(x) \sum_{y \in Y} p(y \mid x) f_j(x,y) \qquad = \sum_{x \in S} \tilde{p}(x) \sum_{y \in Y} p(y \mid x) f_j(x,y)$$

$$= \frac{1}{N} \sum_{i=1}^{N} \sum_{y \in Y} p(y \mid x_i) f_j(x_i, y)$$

An example

- Suppose $P(y \mid x_i) = 1/3$
- Training data:

x1		t1 t2 t3
x2		t1 t4
x3		t4
x4		t1 t3

"Raw" counts

	t1	t2	t3	t4
c1	3/3	1/3	2/3	2/3
c2	3/3	1/3	2/3	2/3
c3	3/3	1/3	2/3	2/3

$$E_p f_j = \frac{1}{N} \boxed{\sum_{i=1}^{N} \sum_{y \in Y} p(y \mid x_i) f_j(x_i, y)}$$

An example

- Suppose $P(y|x_i) = 1/3$
- Training data:

x1	t1	t2	t3
x2	t1	t4	
x3	t4		
x4	t1	t3	

Model expectation

	t1	t2	t3	t4
c1	3/12	1/12	2/12	2/12
c2	3/12	1/12	2/12	2/12
c3	3/12	1/12	2/12	2/12

$$E_p f_j = \frac{1}{N} \sum_{i=1}^{N} \sum_{y \in Y} p(y \mid x_i) f_j(x_i, y)$$

Calculating model expectation

$$E_p f_j = \frac{1}{N} \sum_{i=1}^{N} \sum_{y \in Y} p(y \mid x_i) f_j(x_i, y)$$

Let N be the number of training instances

for each instance x in the training data

  calculate P(y|x) for every $y \in Y$

  for each feature t in x

    for each $y \in Y$

      model_expect [t] [y] += 1/N * P(y | x)

Empirical expectation vs. model expectation

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

$$E_p f_j = \frac{1}{N} \sum_{i=1}^{N} \boxed{\sum_{y \in Y}} \boxed{p(y \mid x_i)} f_j(x_i, \boxed{y})$$

Outline for modeling

- Feature function: $f_j(x, y)$

- Calculating the expectation of a feature function

- The forms of P(x,y) and P(y | x)**

Constraints

- Model expectation = Empirical expectation

$$E_p f_j = E_{\tilde{p}} f_j = d_j$$

- Why impose such constraints?
  - The MaxEnt principle: Model what is known
  - To maximize the conditional likelihood: see Slides #24-28 in (Klein and Manning, 2003)

The conditional likelihood (**)

- Given the data (X,Y), the conditional likelihood is a function of the parameters λ

$$logP(Y|X,\lambda)$$

$$= log\prod_{(x,y)\in(X,Y)} P(y|x,\lambda)$$

$$= \sum_{(x,y)\in(X,Y)} logP(y|x,\lambda)$$

$$= \sum_{(x,y)\in(X,Y)} log\frac{e^{\sum_j \lambda_j f_j(x,y)}}{\sum_{y\in Y} e^{\sum_j \lambda_j f_j(x,y)}}$$

$$= \sum_{(x,y)\in(X,Y)}(log\ e^{\sum_j \lambda_j f_j(x,y)} - log\sum_{y\in Y}\ e^{\sum_j \lambda_j f_j(x,y)})$$

$$= ...$$

The effect of adding constraints

- Bring the distribution closer to the data
- Bring the distribution further away from uniform
- Lower the entropy
- Raise the likelihood of data

Restating the problem

The task: find p* s.t. $$p^* = \arg\max_{p \in P} H(p)$$

where $$P = \{ p \mid E_p f_j = E_{\tilde{p}} f_j, j = \{1, \ldots, k\} \}$$

Objective function: H(p)

Constraints: $$\{ E_p f_j = E_{\tilde{p}} f_j = d_j, j = \{1, \ldots, k\} \}$$

Using Lagrange multipliers (**)

Minimize A(p):

$$A(p) = -H(p) - \sum_{j=1}^{k} \lambda_j (E_p f_j - d_j) - \lambda_0 (\sum_{x,y} p(x,y) - 1)$$

$$A'(p) = 0$$

$$\Rightarrow \frac{\delta(\sum_{x,y} p(x,y) \log p(x,y) - \sum_{j=1}^{k} \lambda_j ((\sum_{x,y} p(x,y) f_j(x,y)) - d_j) - \lambda_0 (\sum_{x,y} p(x,y) - 1))}{\delta p(x,y)} = 0$$

$$\Rightarrow 1 + \log p(x,y) - \sum_{j=1}^{k} \lambda_j f_j(x,y) - \lambda_0 = 0$$

$$\Rightarrow \log p(x,y) = (\sum_{j=1}^{k} \lambda_j f_j(x,y)) + \lambda_0 - 1$$

$$\Rightarrow p(x,y) = e^{\sum_{j=1}^{k} \lambda_j f_j(x,y) + \lambda_0 - 1} = e^{\sum_{j=1}^{k} \lambda_j f_j(x,y) + \lambda_0 - 1}$$

$$\Rightarrow p(x,y) = \frac{e^{\sum_{j=1}^{k} \lambda_j f_j(x,y)}}{Z} \quad where\ Z = e^{1-\lambda_0}$$

Questions

$$p^* = \arg\max_{p \in P} H(p)$$

where $$P = \{ p \mid E_p f_j = E_{\tilde{p}} f_j ,\ j = \{1,...,k\} \}$$

- Is P empty?
- Does p* exist?
- Is p* unique?
- What is the form of p*?
- How to find p*?

What is the form of p*? (Ratnaparkhi, 1997)

$$P = \{ p \mid E_p f_j = E_{\tilde{p}} f_j, j = \{1,...,k\} \}$$

$$Q = \{ p \mid p(x,y) = \pi \prod_{j=1}^{k} \alpha_j^{f_j(x,y)}, \alpha_j > 0 \}$$

Theorem: if $p^* \in P \cap Q$ then $p^* = \arg\max_{p \in P} H(p)$

Furthermore, p* is unique.

Two equivalent forms

$$p(x,y) = \pi \prod_{j=1}^{k} \alpha_j^{\,f_j(x,y)}$$

$$p(x,y) = \frac{e^{\sum_{j=1}^{k} \lambda_j f_j(x,y)}}{Z}$$

$$\Rightarrow \quad \pi = \frac{1}{Z} \qquad \lambda_j = \ln \alpha_j$$

Modeling summary

Goal: find p* in P, which maximizes H(p).

$$P = \{ p \mid E_p f_j = E_{\tilde{p}} f_j, j = \{1,...,k\} \}$$

It can be proved that, **when** p* exists

- it is unique
- it maximizes the conditional likelihood of the training data
- it is a model in Q, where

$$Q = \{ p \mid p(x) = \pi \prod_{j=1}^{k} \alpha_j^{f_j(x)}, \alpha_j > 0 \}$$

Outline

- Overview
- The Maximum Entropy Principle

- Modeling**
- Decoding

- Training**

- Case study: POS tagging

Decoding

Decoding

$$p(y \mid x) = \frac{e^{\sum_{j=1}^{k} \lambda_j f_j(x,y)}}{Z}$$

Z is the normalizer.

	t_1	t_2	…	t_k
c_1	λ_1	λ_2	…	λ_k
c_2	…	…	…	…
…	…			
c_i	…			λ_{ki}

The procedure for calculating P(y | x)

```
Z=0;
for each y ∈ Y
   sum = 0;   // or sum = default_weight_for_class_y;
   for each feature t present in x
       sum += the weight for (t, y);
   result[y] = exp(sum);
   Z += result[y];

for each y ∈ Y
   P(y|x) = result[y] / Z;
```

MaxEnt summary so far

- Idea: choose the p* that maximizes entropy while satisfying all the constraints.
- p* is also the model within a model family that maximizes the conditional likelihood of the training data.
- MaxEnt handles overlapping features well.
- In general, MaxEnt achieves good performances on many NLP tasks.
- Next: Training: many methods (e.g., GIS, IIS, L-BFGS).

Hw5

Q1: run Mallet MaxEnt learner

- The format of the model file:

FEATURES FOR CLASS c1

<default> 0.3243

t1 0.245

t2 0.491

….

FEATURES FOR CLASS c2

<default> 0.3243

t1 -30.412

t2 1.349

….

Q2: write a MaxEnt decoder

The formula for P(y|x):

$$p(y|x) = \frac{e^{\lambda_0(y) + \sum_{k=1}^{K} \lambda_k f_k(x,y)}}{Z}$$

$\lambda_0(y)$ is the weight of the default feature for y.

The k in f_k corresponds to a (class, feature) pair (c_i, t_j)

$f_k(x, y) = 1$ iff t_j is present in x and $y = c_i$.

Q2: calculate P(y|x)

- The format of the model file:

FEATURES FOR CLASS c1

<default> 0.324

t1 0.245

t2 0.491

t3 -0.22

FEATURES FOR CLASS c2

<default> 0.456

t1 -30.4

t2 1.349

t3 2.42

Suppose x is "t1 t3"

$$p(c_1|x) = \frac{e^{\lambda_0(c_1) + \sum_{k=1}^{K} \lambda_k f_k(x, c_1)}}{Z}$$

$$p(c_1|x) = \frac{e^{0.324+0.245-0.22}}{Z}$$

$$p(c_2|x) = \frac{e^{\lambda_0(c_2) + \sum_{k=1}^{K} \lambda_k f_k(x, c_2)}}{Z}$$

$$p(c_2|x) = \frac{e^{0.456-30.4+2.42}}{Z}$$

$$P(c1 \mid x) = \frac{A}{Z}$$

$$P(c2 \mid x) = \frac{B}{Z}$$

$$P(c3 \mid x) = \frac{C}{Z}$$

$$\mathrm{Z} = \mathrm{A} + \mathrm{B} + \mathrm{C}$$

$$P(c1 \mid x) = \frac{A}{A+B+C}$$

$$P(c2 \mid x) = \frac{B}{A+B+C}$$

$$P(c3 \mid x) = \frac{C}{A+B+C}$$

- train2.vectors.txt
- train2.vectors

- test2.vectors.txt
- test2.vectors

info2vectors –input test2.vectors.txt –output test2.vectors –use-pipe-from train2.vectors

Q3-Q4: calculate expectation

$$E_{\tilde{p}} f_j = \sum_{(x,y)\in X\times Y} \tilde{p}(x,y) f_j(x,y)$$

$$E_p f_j = \sum_{(x,y)\in X\times Y} p(x,y) f_j(x,y)$$