MEM

(Maximum Entropy Model)

Wang Houfeng

Institute of Computational Linguistics

Peking University

Outline

- **Introduction**
- Feature based Model
- Maximum Entropy Models
- Smoothing
- NLP Applications

Classification

- Linear classification:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = \sum w_i x_i + b$$

$$\text{c= sgn}\left(\mathbf{w}^T\mathbf{x} + b\right) = \begin{cases} +1 & if \quad \langle \mathbf{w} \cdot \mathbf{x} \rangle + b > 0 \\ -1 & otherwise \end{cases}$$

- Probability classification

$$c = \arg\max_{c \in C} P(c \mid \mathbf{x})$$

➢How to compute the probability?

Example-1: Part of Speech (POS)

- 在： (三个词性)

  p： 放在桌上

  v: 他在家里

  ad: 他在写作业

- There exists a model p with constraint:

  $P(p \mid 在) + P(v \mid 在) + P (ad \mid 在) = 1$

Example-1(cont)

- If we have one more constraint observed from training data: P(p|在)=2/3
- What about P(v | 在)=? and P (ad |在)=?

  under P(v | 在) + P (ad |在) = 1/3

Example-1(cont)

- infinite possible cases:

  case-1: P(v | 在)=1/6, P (ad |在) =1/6

  case-2: P(v | 在)=2/9, P (ad |在) =1/9

  case-3: P(v | 在)=3/15, P (ad |在) =2/15

……

How to find the model to compute the optimal probability?

Example-2: Machine Translation

- Take (x) into:
  - C1: 抓住
  - C2: 拿走
  - C3: 乘车
  - C4: 测量（体温）
  - C5: 装
  - C6: 花费
  - C7: 理解

$$P(c1|x)+ P(c2|x)+ P(c3|x)+ P(c4|x)+ P(c5|x)+ P(c6|x)+ P(c7|x) = 1$$ (Model P with constraint)

Example-2 (cont.)

- One more constraint observed from training data:

  $P(t1|x)+P(t2|x)=2/5$;

  how about:

$P(t3|x)$, $P(t4|x)$, $P(t5|x)$,$P(t6|x)$,$P(t7|x)$?

Too many cases satisfy the constraint!

Which model is the best one

Outline

- Introduction
- **Feature based Model**
- Maximum Entropy Models
- Smoothing
- NLP Applications

Feature

- Features are elementary pieces of evidence that link aspects of what we observe context **x** and a category y we want to predict.
- A feature f has a real value: f: X×Y→R, where, Y: a set of categories; X: a set of contexts and R: real set.
- In general, feature is indicator function{0/1}.

$$f_i(x, y) \equiv [\Phi(x) \wedge y = y_i]$$

Feature

- Why feature
  - Feature tell which factors have effect on making decision (disambiguation/classification)
  - Context of a point usually be considered in NLP
- Weight of feature
  - Stands for important degree of the feature.
- Example:
  - For linear discrimination:

$f(x)=w_1x_1+w_2x_2+\ldots+w_nx_n+b$

Typical Models

- Text Categorization
- Example: (data)
  - Label: Finance(business)
  - Input: 近年中国股市连续走牛是否意味房价走跌？
  - Features:

{中国,股市,走牛,房价,走跌}

Typical Models

- WSD(word sense disambiguation)
- Example: “冲”
  - Label: 毁坏,卷走
  - Input: 洪水 冲 走了房屋
  - Features:

{w_{-1}=洪水， w_1=走， w_2=房屋}

Typical Models

- POS Tagging
- Example: “把”
  - Label: p
  - Input:他 把 书 放回 到 了 原处
  - Features:

{w_0=把， w_{-1}=他， T_{-1}=r, w_1 =书}

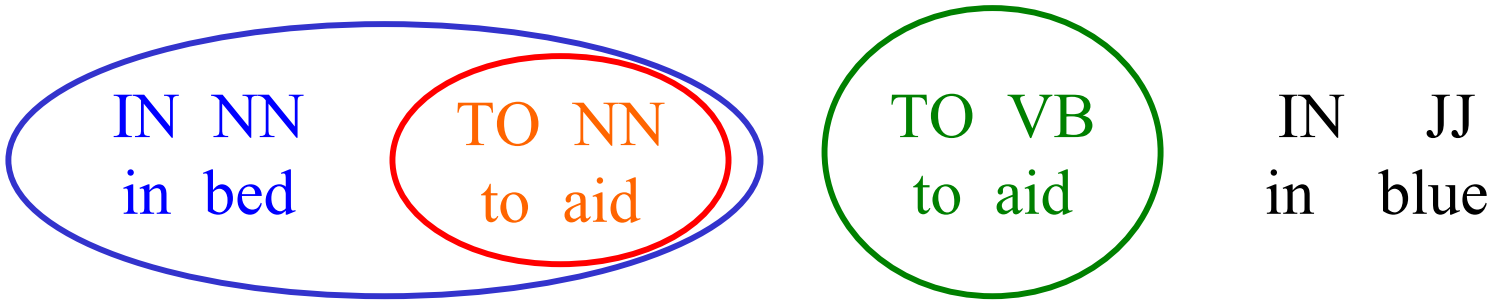
Feature

- We usually say Φ(x) is a feature of data x, at the same , Φ(x) ∧y=y_i is a feature of class-data pair (x,y).
- For example in English:

$$f_1(x,y) = [y = "NN" \wedge islower(w_0) \wedge ends(w_0, "d")]$$

$$f_2(x,y) = [y = "NN" \wedge w_{-1} = "to" \wedge t_{-1} = "TO"]$$

$$f_3(x,y) = [y = "VB" \wedge islower(w_0)]$$



Feature

- For Tagging: features can include:
  - Current, previous, next words in isolation or together;
  - Previous one , two or three tags
  - Other features: suffixes for English

-3	-2	-1	0	1
DT	NNP	VBD	**??**	??
The	Dow	fell	**22.6**	%

feature

w_0	**22.6**
w_{-1}	Fell
w_1	%
T_{-1}	VBD
$T_{-2}\ T_{-1}$	NNP-VBD
Hasdigit?	true
…	…

Feature-based Classifiers

- Classifier:
  - Classify from feature Sets $\{f_i\}$ to classes $\{y\}$;
  - Assign a weight (importance measure) λ_i to feature f_i ;

$$f_1(x,y) = [y = "NN" \wedge islower(w_0) \wedge ends(w_0, "d")]$$

$$f_2(x,y) = [y = "NN" \wedge w_{-1} = "to" \wedge t_{-1} = "TO"]$$

$$f_3(x,y) = [y = "VB" \wedge islower(w_0)]$$

Feature-based Classifiers

- Exponential(Loglinear, Maxent, Logistic) Model

$$p(y \mid x) = \frac{\exp\{\sum_i \lambda_i f_i(x, y)\}}{\sum_y \exp\{\sum_i \lambda_i f_i(x, y)\}}$$

$$p(NN \mid aid) = e^{1.2}e^{-1.8} \Big/ (e^{1.2}e^{-1.8} + e^{0.3}) = 0.29$$

$$p(VB \mid aid) = e^{0.3} \Big/ (e^{1.2}e^{-1.8} + e^{0.3}) = 0.71$$

Feature-based Classifiers

- The Exponential Model is a way of deciding how to weight each feature, given training data;
- It constructs not only classifications, but probability distribution over classifications;

Outline

- Introduction
- Feature based Model
- ➢ **Maximum Entropy Models**
- Smoothing
- NLP Applications

Jaynes

- Jaynes had said:

  Maximum Entropy agrees with everything that is known, but carefully avoid anything that is unknown

- In other words:

  given a collection of facts, choose a model which is consistent with all facts, but otherwise as uniform as possible

Modeling the problem

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

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

Maximun Entropy Distribution

A distribution $\mathbf{p}=\{p_1, p_2, ..., p_n\}$ on a finite set $\{1, 2, ...n\}$

Its entropy is defined as $H(\mathbf{p}) = -\sum_{i=1}^{n} p_i \log p_i$, where, $\sum_{i=1}^{n} p_i = 1$;

Maxmize $H(\mathbf{p})$

subject to $-\sum_{i=1}^{n} p_i = 1$

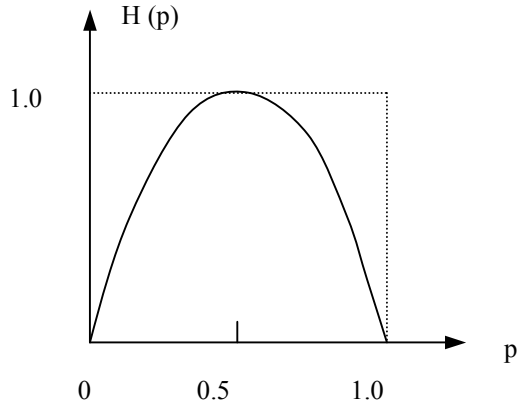
According to Lagrangian,

$$L(\mathbf{p},\beta)=\sum_{i=1}^{n} p_i \log p_i + \beta\left(\sum_{i=1}^{n} p_i - 1\right)$$

$$\frac{\partial L(p,\beta)}{\partial p_i} = \log e p_i + \beta = 0$$

combining $\log e p_i + \beta = 0$ with $\sum_{i=1}^{n} p_i = 1$

$$\Rightarrow p_i = \frac{1}{n}$$



For 2 classes

Conditional Entropy

- Entropy $H\left(p(X)\right) = -\sum_{i=1}^{k} p\left(x = x_i\right) \log p\left(x = x_i\right)$
- Conditional Entropy

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

- Maximum Conditional Entropy

$$\text{Maximize} \quad H\left(p(Y \mid X)\right)$$

Constraint

- Model should be consistent with all facts, but otherwise as uniform as possible;
- What is fact?
  - Training set (or observed data, or event set)
- How to represent fact (each event)?
  - Features!

Expectation of Feature

- Observed(empirical) expectation of a feature:

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

- Model (p) expectation of a feature:

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

$$\approx \sum_{x, y \in X \times Y} \tilde{p}(x) p(y \mid x) f_i(x, y)$$

Maximum Entropy(shannon)

$$\max_{P} \; H(P(Y \mid X)) = -\sum_{x,y} p(x,y) \log p(y \mid x)$$

$$\approx -\sum_{x,y} \tilde{p}(x)\, p(y \mid x) \log p(y \mid x) \qquad .....(1)$$

subject to $(s.t)$

$$\forall i, \; \sum_{x,y} \tilde{p}(x,y)\, f_i(x,y) = \sum_{x,y} \tilde{p}(x)\, p(y \mid x)\, f_i(x,y) \quad ....(2)$$

$$\forall x, \; \sum_{y} p(y \mid x) = 1 \qquad .....(3)$$

$$\forall x, y, \; p(y \mid x) \geq 0 \qquad .....(4)$$

Maximum Entropy

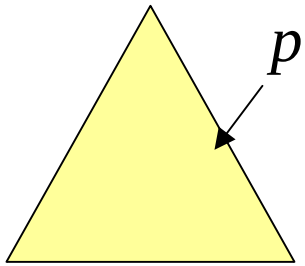
$$p^* = \underset{p \in P}{\operatorname{argmax}}\ H(p) \quad \text{where} \quad P = \{p \mid E_{\tilde{p}} f_i = E_p f_i\}$$

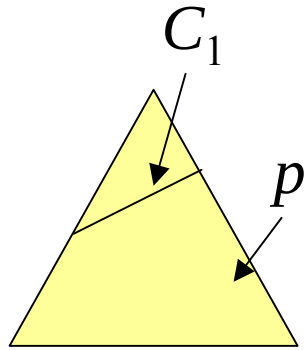
where

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

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

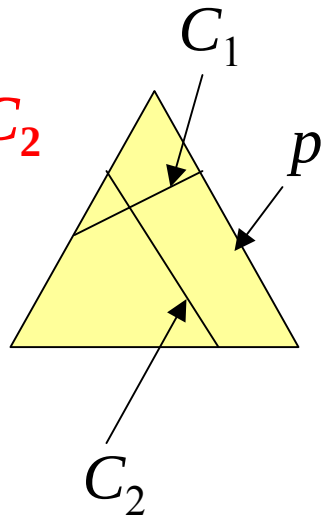
No constraints
All p∈P is allowable

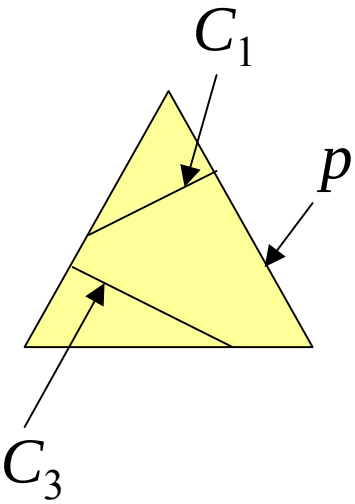




p∈ C_1

p∈ $C_1 \cap C_2$

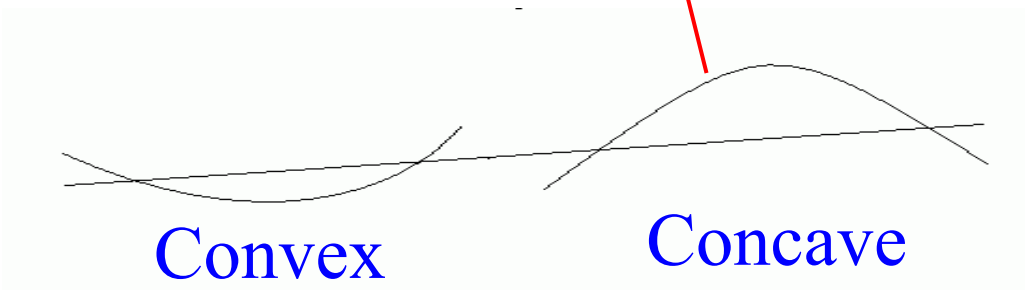




No solutions

Concave- Constrained Optimization

- H(p) is a ∩-function => maximum value
- constraint optimization problem
  - Lagrangian
  - Dual form



Lagrangian Multiplier

$$L(P,\Lambda,\alpha) = H(P)$$
$$+\sum_i \lambda_i \{\sum_{x,y} \tilde{p}(x)\, p(y \mid x)\, f_i(x,y) - \sum_{x,y} \tilde{p}(x,y)\, f_i(x,y)\}$$
$$+\sum_x \alpha_x \{\sum_y p(y \mid x) - 1\} \qquad \text{.....(5)}$$

- Can think of unconstraint optimization as (Max-Min):

$$\max_{P(y|x)} \min_{\lambda} L(P,\Lambda,\alpha)$$

or, duality let us reverse the ordering:

$$\min_{\lambda} \max_{P(y|x)} L(P,\Lambda,\alpha)$$

Cont.

- **Dual method** works in this way:
  - solve the maximization for a given set of λs;
  - then, of above solution, minimize over the space of λs

Where, $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_n\}$. Now,hold λ_i fixed, maximize $L(p, \Lambda, \alpha)$

$$\frac{\partial L(P,\Lambda,\alpha)}{\partial p(y|x)} = -\tilde{p}(x)\{\log P(y|x) + p(y|x)\frac{1}{p(y|x)}\} + \sum_i \lambda_i \tilde{p}(x) f_{i(x,y)} + \alpha_x = 0$$

$$\Rightarrow \quad p(y|x) = \exp\{\sum_i \lambda_i f_i(x,y) - 1 + \frac{\alpha_x}{\tilde{p}(x)}\} = \exp\{\sum_i \lambda_i f_i(x,y)\} \exp\{\frac{\alpha_x}{\tilde{p}(x)} - 1\}$$

(cont.)

Since $\sum_y p(y \mid x) = 1$,

$$\sum_{\mathrm{y}} \exp\{\frac{\alpha_x}{\tilde{p}(x)} - 1\} \exp\{\sum_i \lambda_i f_i(x, y)\} = 1$$

$$\Rightarrow \quad \exp\{\frac{\alpha_x}{\tilde{p}(x)} - 1\} \sum_{\mathrm{y}} \exp\{\sum_i \lambda_i f_i(x, y)\} = 1$$

$$\Rightarrow \quad \exp\{\frac{\alpha_x}{\tilde{p}(x)} - 1\} = \frac{1}{\sum_y \exp\{\sum_i \lambda_i f_i(x, y)\}}$$

Therefore,

$$\mathrm{p}(y \mid x) = \frac{1}{\sum_y \exp\{\sum_i \lambda_i f_i(x, y)\}} \exp\{\sum_i \lambda_i f_i(x, y)\} \qquad \text{.....(6)}$$

Z(x)

(cont.)

Let:

$$Z(x) = \exp(\frac{a_x}{\tilde{p}(x)} - 1) = \frac{1}{\sum_y \exp\{\sum_i \lambda_i f_i(x, y)\}}$$

$$\Rightarrow \quad p(y \mid x) = Z(x) \exp\{\sum_i \lambda_i f_i(x, y)\} \quad .....(7)$$

This task seems to find $\Lambda = \{\lambda_1, \lambda_2, ... \lambda_n\}$!

Substitute above p(y|x) into H(P)

$$
\begin{aligned}
H(P) &= -\sum_{x,y} \tilde{p}(x)\, p(y \mid x) \log p(y \mid x) \\
&= -\sum_{x,y} \tilde{p}(x) \frac{\exp\{\sum_i \lambda_i f_i(x,y)\}}{\sum_y \exp\{\sum_i \lambda_i f_i(x,y)\}} \log \frac{\exp\{\sum_i \lambda_i f_i(x,y)\}}{\sum_y \exp\{\sum_i \lambda_i f_i(x,y)\}} \\
&= -\sum_{x,y} \tilde{p}(x) \frac{\exp\{\sum_i \lambda_i f_i(x,y)\}}{\sum_y \exp\{\sum_i \lambda_i f_i(x,y)\}} (\sum_i \lambda_i f_i(x,y) - \log \sum_y \exp\{\sum_i \lambda_i f_i(x,y)\}) \\
&= -\sum_{x,y} \tilde{p}(x) \frac{\exp\{\sum_i \lambda_i f_i(x,y)\}}{\sum_y \exp\{\sum_i \lambda_i f_i(x,y)\}} \sum_i \lambda_i f_i(x,y) \\
&\quad + \sum_x \tilde{p}(x) \log \sum_y \exp\{\sum_i \lambda_i f_i(x,y)\}) \qquad (8)
\end{aligned}
$$

Substitute above P(y|x) and H(P) into L(p*,Λ,α*)

$$\Psi(\Lambda) = L(P^*, \Lambda, \alpha^*) = -\sum_{x,y} \tilde{p}(x) \frac{\exp\{\sum_i \lambda_i f_i(x,y)\}}{\sum_y \exp\{\sum_i \lambda_i f_i(x,y)\}} \sum_i \lambda_i f_i(x,y)$$

$$+ \sum_x \tilde{p}(x) \log \sum_y \exp\{\sum_i \lambda_i f_i(x,y)\}$$

$$+ \sum_i \lambda_i \{\sum_{x,y} \tilde{p}(x) p(y \mid x) f_i(x,y) - \sum_{x,y} \tilde{p}(x,y) f_i(x,y)\}$$

$$= \sum_x \tilde{p}(x) \log \sum_y \exp\{\sum_i \lambda_i f_i(x,y)\} - \sum_i \lambda_i \sum_{x,y} \tilde{p}(x,y) f_i(x,y)$$

$$= \sum_x \tilde{p}(x) \log Z(x) - \sum_i \lambda_i \tilde{p}(f_i) \qquad (9)$$

$$\min_{\Lambda} \Psi(\Lambda)$$

$$Z(x) = \frac{1}{\sum_y \exp\{\sum_i \lambda_i f_i(x,y)\}}$$

Primal vs. Dual

Primal

$$\max_{P} \; H(P(Y \mid X)) = -\sum_{x,y} \tilde{p}(x)\, p(y \mid x) \log p(y \mid x)$$

$$subject\ to\ (s.t)$$

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

$$\forall x,\ \sum_{y} p(y \mid x) = 1 \quad \text{and} \quad \forall x, y,\ p(y \mid x) \geq 0$$

Constrained optimization

Dual

$$\min_{\Lambda} \; \Psi(\Lambda) = \sum_{x} \tilde{p}(x) \log Z(x) - \sum_{i} \lambda_i \tilde{p}(f_i)$$

Unconstrained optimization

$$Z(x) = \frac{1}{\sum_{y} \exp\{\sum_{i} \lambda_i f_i(x,y)\}}$$

Relation to Maximal Likelihood

- Assuming
  - A coin has a probability p of heads, 1-p of tails.
  - Observation: We toss a coin N times, and the result is a set of Hs and Ts, and there are M Hs.
- What is the value of p based on MLE, given the observation?
  - The Likelihood is defined as: $L(p) = p^M (1-p)^{N-M}$

Log-likelihood is:

$$Log\_L(p) = M \log p + (N-M)\log(1-p)$$

If Likelihook has a maximum, its log also has at the same point!

Relation to Maximal Likelihood

- For conditional probability P(y|x), the log-likelihood is:

$$Log\_L(P) = \sum_{x,y} C(x,y) \log p(y \mid x) = N \sum_{x,y} \tilde{p}(x,y) \log p(y \mid x)$$

where, N: is total of events, and $\sum_{x,y} \tilde{p}(x,y) \log p(y \mid x)$

$$= \sum_{x,y} \tilde{p}(x,y) \{ \sum_{i} \lambda_i f_i(x,y) - \log \sum_{y} \exp(\sum_{i} \lambda_i f_i(x,y)) \}$$

$$= \sum_{i} \lambda_i \tilde{p}(f_i) - \sum_{x,y} \tilde{p}(x,y) \log \sum_{y} \exp(\sum_{i} \lambda_i f_i(x,y))$$

$$= \sum_{i} \lambda_i \tilde{p}(f_i) - \sum_{x,y} \tilde{p}(x) \log \sum_{y} \exp(\sum_{i} \lambda_i f_i(x,y))$$

$$= - L(P^*, \Lambda, a^*) \qquad (10)$$

Therefore, $\max_{\lambda} \text{Log\_L}(P) = \min_{\lambda} L(P^*, \Lambda, a^*)$

Compute Parameters for Model

- Find λ^*, such that $\lambda^* = \text{Min } \Psi(\lambda)$;
- It cannot be found analytically;
- An **optimization method** to the maximum entropy problem is the iterative scaling algorithm of Darroch and Ratliff (**GIS**)

Build a Maximum Entropy Model

- Define features templates.
- Extract events from training corpus to form sample set (event set).
- Construct feature set over events.
- Train maximum entropy model(estimate parameters of model).
- Apply model to practical system.

Compute Parameters for Model

Training Set $\Omega = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(N)}, y^{(N)})\}$

$= \{e^{(1)}, e^{(2)}, ..., e^{(N)}\}$

Feature Set $F = \{f_1, f_2, ..., f_n\}$,

Reference distribution (or empirical distribution) $\tilde{p}$:

$$\tilde{p}(e) = \tilde{p}(x, y) = \frac{c(x, y)}{N} = \frac{\sum_{1 \le i \le N} \delta(e, e^{(i)})}{N}$$

where $\delta(e, e^{(i)}) = 1$ *iff* $e = e^{(i)}$

Compute parameters: $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$:

Train a set of parameters, so that, each parameter(λ_j, 1<j<n) corresponding to a only feature(f_j, 1<j<n) in F.

GIS Algorithm

- GIS(Generalized Iterative Scaling) algorithm is used to find values for the parameters of maximum entropy
- GIS procedure **requires that features sum to a constant** for any event*(x,y)* in training data:

$$\sum_{j=1}^{n} f_j(x, y) = C \qquad (11)$$

GIS Algorithm (cont.)

If it is not always true that features sum to a constant for any event*(x,y)* in training data

Then Let $C = \text{Max} \Sigma f_i(x, y)$ and add a correction feature $f_{n+1}(x,y)$ for event(x,y), satisfy:

$$f_{n+1}(x, y) = C - \sum_{j=1}^{n} f_j(x, y) \qquad (12)$$

$$\text{maybe} \quad f_{n+1}(x, y) > 1$$

GIS (Iteration)

1) Initializtion: $\lambda_j^{(0)} = 0$

2) Repeat:

$$\lambda_j^{(m+1)} = \lambda_j^{(m)} + \Delta_j^{(m)}$$

$$where \quad \Delta_j^{(m)} = \frac{1}{c}\log[\frac{E_{\tilde{p}} f_j}{E_{p^{(m)}} f_j}] \qquad (13)$$

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

unchangeable

$$E_{p^{(m)}} f_j = \sum_{x,y} \tilde{p}(x) p^{(m)}(y \mid x) f_j(x,y)$$

$$p^{(m)}(y \mid x) = Z(x) \exp \sum_{j=1}^{n+1} \lambda_j^{(m)} f_j(x,y)$$

altered

Terminal Condition of Iteration

- Cond-1: A fixed number of iterations (200)
- Cond-2: Kullback-Leibler divergence

$$| D(\tilde{p} \,||\, p^{(m)}) = \sum_{x,y} \tilde{p}(x,y) \log \frac{\tilde{p}(x,y)}{p^{(m)}(x,y)} | < \varepsilon$$

or

$$| L(p^{(m)}) - L(p^{(m-1)}) | < \varepsilon$$

Log_Likelihood

On Correction feature

- Obviously, (13) shows that the smaller C is , the bigger the rate of convergence of the algorithm is, i.e. it is preferable to keep C as small as possible.
- While C is required a constant(12), James R. Curran[5] had shown that GIS, like IIS converges to maximum likelihood model without correction feature.

Outline

- Introduction
- Feature based Model
- Maximum Entropy Models
- **Overfitting solution & Smoothing**
- NLP Applications

Too many features

- Lots of features
  - ManEnt Model in NLP might have over 1M feature;
  - High memory cost!
- Lots of Sparsity
  - Many features seen in training set will never occur in future test;
  - Overfitting easy!— need smoothing;
- Optimization problem
  - Feature weights can be infinite! And iterative solver will need to take a long time to get those infinite weights

Solutions to Overfitting

- Feature select: throw out rare features.
  - Require every feature to occur > 4 (threshold) times,
  - It is very easy!
- Feature Induction
  - Keep 1000(threshold) features
  - always greedily picking the ones that most improves performance;
- Smoothing
  - Smooth the observed feature counts
  - Smooth the feature weights by using a prior

Feature Selection

- Select useful feature subset F from universal set U.
- Simple selection strategy:

$$F = \{ f \mid \sum_{(x,y)} f_i(x,y) > threshold \}$$

- Strength:

  simple, and no computational expensive.
- weakness:
  - Many selected feature is redundant;
  - The burden falls into parameter estimation

Complicated Feature selection:Feature Induction

$Initial\ Data:$

$Empirical\ Distribution:\ \tilde{P}$

$Feature\ S = \phi$

$Thus,\ P_S\ is\ uniform(condition\ probability?)$

$Output:$

$A\ feature\ set\ S = \{f_1, f_2, ..., f_m\},\ and$

$$\underset{P_S}{ArgMin}\, D(\tilde{P} \,||\, P_S)$$

Feature Induction Algorithm

1. Do for each candidate feature $f \in F$:

   Compute the model $P_{S \cup \{f\}}$ Using GIS algorithm;

   Compute $Gain(f) = D(\tilde{P} \,||\, P_S) - D(\tilde{P} \,||\, P_{S \cup \{f\}})$

   or Compute $Gain(f) = L(P_{S \cup \{f\}}) - L(P_S)$

2. If termination condition apply, then exit.

3. select the feature $\hat{f} = \underset{f \in F}{Max}\ Gain(f)$.

4. adjoin $\hat{f}$ to S, and $F = F - \{\hat{f}\}$.

5. Go to step1.

Feature selection Vs Feature induction

- There is not definite conclusion that which strategy is better than the other.
- In practice, count cutoff strategy is widely applied for it is simple.

Smoothing(1): Gaussian Prior

- Estimating the parameters of a ME model by Maximizing objective function: max-likelihood (10) with constraints (2)(3)(4).
- using Gaussian prior penalty to smooth ME model and avoid overfiting the incomplete event space

Cont.

- Let likelihood function (10) is denoted as L(Λ).
- Add Gaussian Prior, our objective function L'(Λ) :

$$L'(\Lambda) = L(\Lambda) + \sum_{1}^{F} \log \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{\lambda_i^2}{\sigma_i^2}\right)$$

$$= L(\Lambda) - \sum_{1}^{F} \frac{\lambda_i^2}{\sigma_i^2} + const(\Lambda)$$

$where\ \sigma_i^2$

are the variances of the gaussian with feature f_i. It can be estimated from empirical distribution of f_i in training data

ME with Gaussian Prior

- L'(Λ) is no longer the likelihood function but a posteriori. The effect of the positive is to penalize models that have large positive or negative weights.
- The incremental values Δ satisfy that:

$$E_{\tilde{p}} f_i = E_{p^{(m)}} f_i \exp(\Delta_i^{(m)} C(x,y)) + \frac{\lambda_i^{(m)} + \Delta_i^{(m)}}{\sigma_i^2} \qquad (14)$$

- The incremental values Δ in (13) are replaced by (14):
- However, (14) does not have an analytic solution for Δ, it can be solved using Newtom-Raphson.
- $C(x,y) = \text{Max} \Sigma f_j(x,y)$

Other Smoothing: exponential Prior

- Iteration in GIS (13) is replaced by:

$$\lambda_i^{(m+1)} = Max\{0, (\lambda_i^{(m)} + \Delta_i^{(m)})\} \qquad (15)$$

$$where, \Delta_i^{(m)} = \frac{1}{C(x,y)} \log \frac{E_{\tilde{p}} f_i - \alpha_i}{E_p f_i})$$

$$\alpha_i = \frac{1}{\sqrt{\sigma_i^2}}$$

$$C(x,y) = Max(\sum_i f_i(x,y))$$

- See reference[6]

Outline

- Introduction
- Feature based Model
- Maximum Entropy Models
- Overfitting solution & Smoothing
- **NLP Applications**

Some applications

- ✓**Word Segmentation**
- POS tagging
- Named entity recognition
- WSD

Word Segmantation

(From **Nianwen Xue**)

Assign a tag to each character in a sentence based on the **p**osition **o**f the **c**haracter within a word (POC tag):

by itself:	产/LR	产	‘produce’
on the left:	产/LL	产品	‘product’
in the middle:	产/MM	生产力	‘productivity’
on the right:	产 RR	投产	'start production'

Ambiguity arises when a character has multiple tags:

日/LL|LR 文/RR|LL 章/LL|RR 鱼/RR|LR 怎/LL 么/RR 说/LR ?

The task then is to pick the correct tag based on the context, in a manner similar to the part-of-speech tagging problem.

More examples

- A Manually Segmented Sentence:

国有 企业 与 外商 投资 企业 齐头并进 ， 国有 企业 继续 居于 主导 地位 ， 外商 投资 企业 仍然 发挥 重要 的 作用 。

- A POC-tagged Sequence Automatically Derived from the Manual Segmentation:

国_LL 有_RR 企_LL 业_RR 与_LR 外_LL 商_RR 投_LL 资_RR 企_LL 业_RR 齐_LL 头_MM 并_MM 进_RR ，_LR 国_LL 有_RR 企_LL 业_RR 继_LL 续_RR 居_LL 于_RR 主_LL 导_RR 地_LL 位_RR ，_LR 外_LL 商_RR 投_LL 资_RR 企_LL 业_RR 仍_LL 然_RR 发_LL 挥_RR 重_LL 要_RR 的_LR 作_LL 用_RR 。_LR

Feature

- Encode contextual information that is useful in predicting the tag of a character with features.
- Examples
  - If the current character is W_i, then it should be tagged T_i
  - If the previous character is tagged T_{i-1}, then the current character should be tagged T_i
  - If the previous character is W_{i-1} and the current character is W_i, then the current character should be tagged T_i
- The features will be instantiations of a pre-defined set of feature templates

Feature templates

i. The current character
ii. The previous (next) character and the current character
iii. The previous (next) two characters
iv. The previous character and the next character
v. The tag of the previous character
vi. The tag of the character two before the current character

Some applications

- Word Segmentation
- ✓**POS tagging**
- Named entity recognition
- WSD

POS tagging
(From Ratnaparkhi)

- Notation:
  - $f_j(h_i, t_i)$: h: history for i^{th} word, t: tag for i^{th} word
- History(or context):

$$h_i = \{w_i, w_{i-1}, w_{i-2}, w_{i+1}, w_{i+2}, t_{i-1}, t_{i-2}\}$$

- Training data:
  - Treat it as a list of (h_i, t_i) pairs.
  - How many pairs are there?

MaxEnt Model for tagging

- Training:
  - Define features templates
  - Create the feature set
  - Determine feature weights via GIS/IIS

$$P(t_1,...,\ t_n \mid w_1,...,\ w_n)$$

$$= \prod_{i=1}^{n} p(t_i \mid w_1^n, t_1^{i-1})$$

$$\approx \prod_{i=1}^{n} p(t_i \mid h_i)$$

Features templates

Condition	Features	
W_i is not rare	W_i =X	& t_i=T
W_i is rare	X is prefix of W_i ,|X|≤ 4	& t_i=T
	X is suffix of W_i ,|X|≤ 4	& t_i=T
	W_i contain number	& t_i=T
	W_i contain uppercase character	& t_i=T
	W_i contain hyphen	& t_i=T
$\forall$ W_i	t_{i-1}=X	& t_i=T
	t_{i-2} t_{i-1} =XY	& t_i=T
	W_{i-1} =X	& t_i=T
	W_{i-2} =X	& t_i=T
	W_{i+1} =X	& t_i=T
	W_{i+2} =X	& t_i=T

History h_i Tag t_i

Feature Set

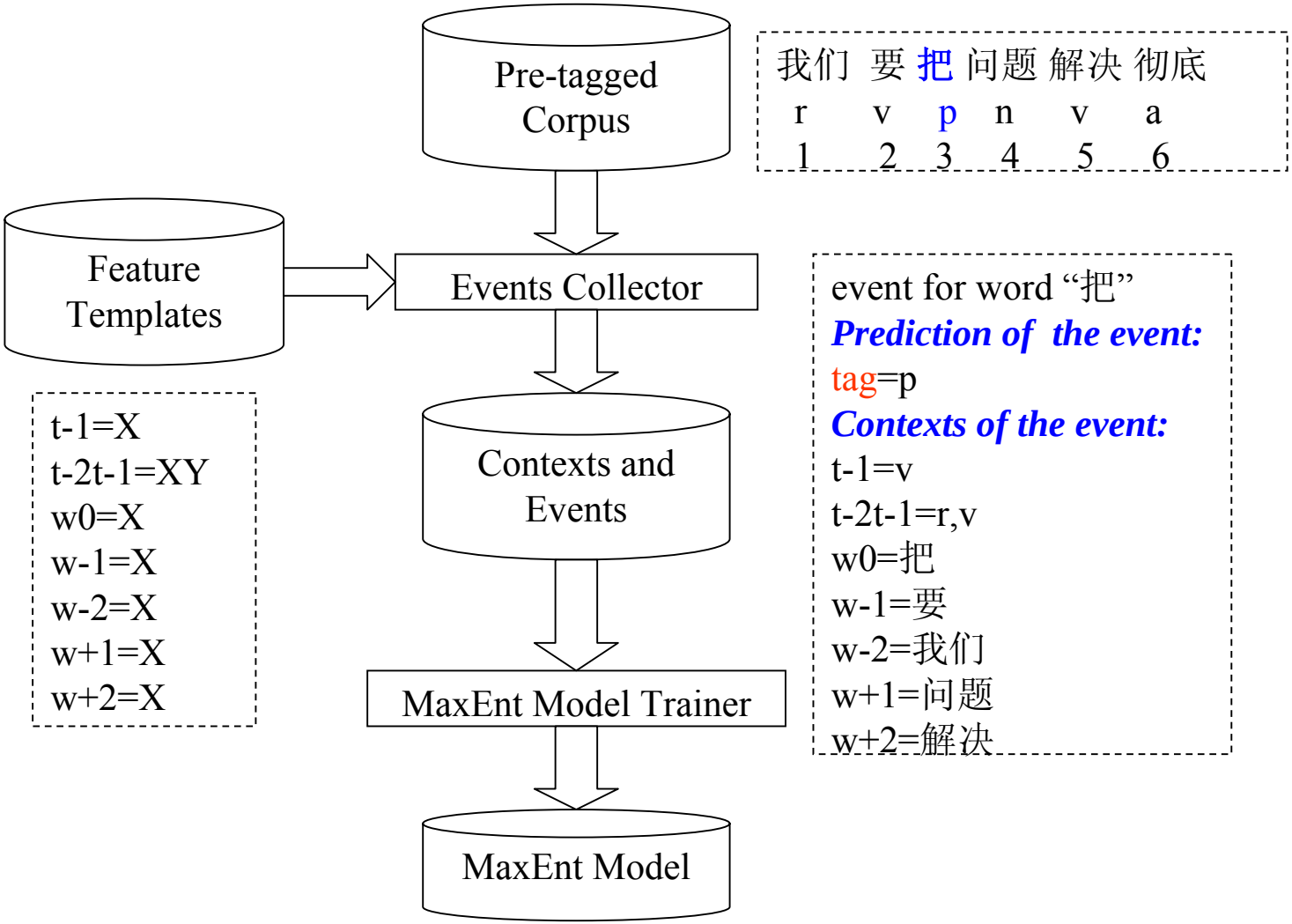
Word	我们	要	把	问题	解决	彻底
Tag	r	v	p	n	v	a
position	1	2	3	4	5	6

W_i =把	& t_i=p
W_{i-1} =要	& t_i= p
W_{i-2} =我们	& t_i= p
W_{i+1} = 问题	& t_i= p
W_{i+2} = 解决	& t_i= p
t_{i-1}=**v**	& t_i= p
t_{i-2} t_{i-1} =**r v**	& t_i= p

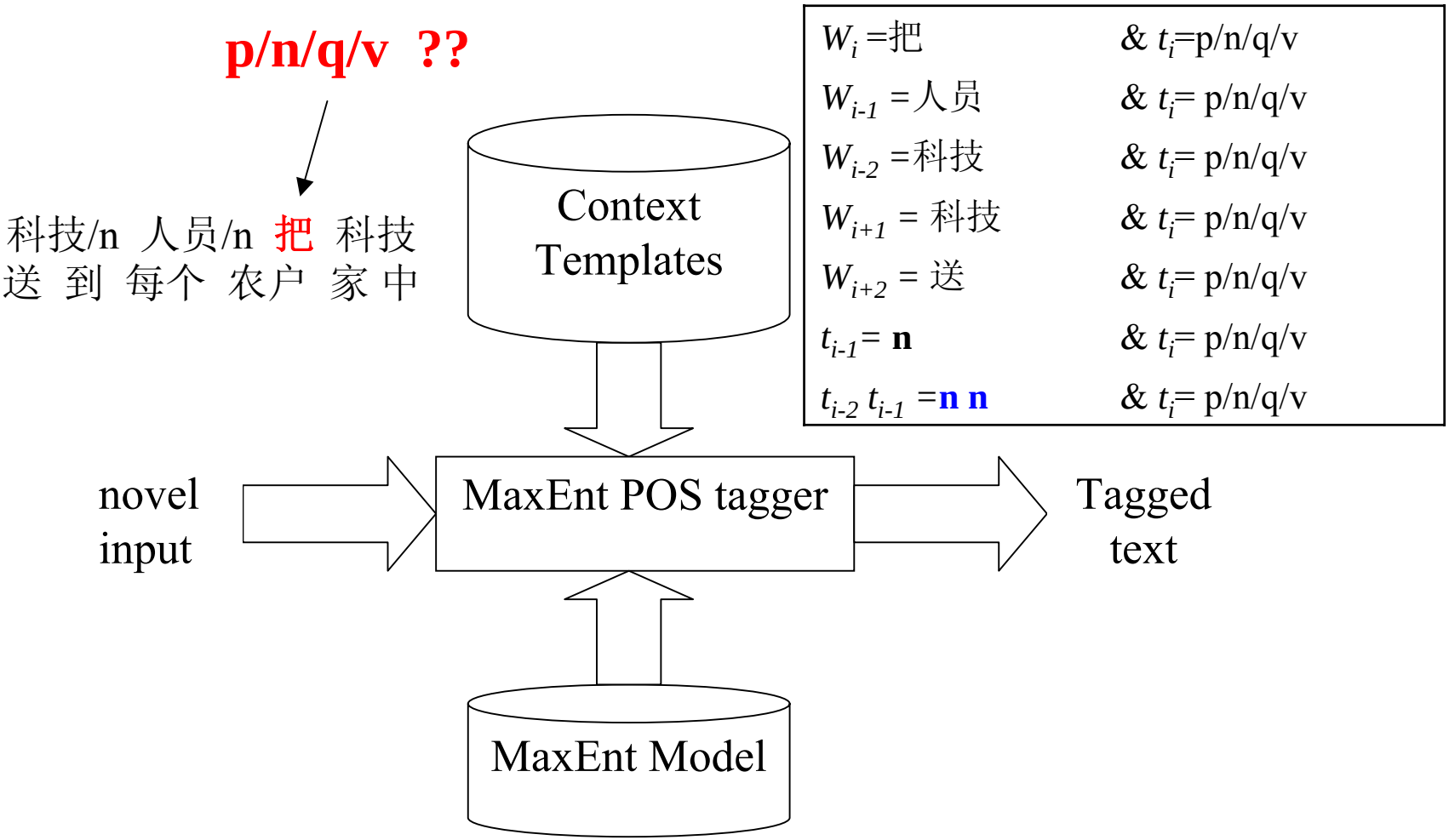
➔ Collect all the features from the training data

➔ Throw away features that appear less than 10 times

Train a Model for a ME tagger



Use ME tagger



Test

- The test corpus is tagged **one sentence at a time**
- The testing procedure requires **a search to list the candidate tag sequences** for the sentence
- The tag sequence with the **highest probability** is chosen as the answer
- The search algorithm is usually a top *K* breadth first search(BFS)

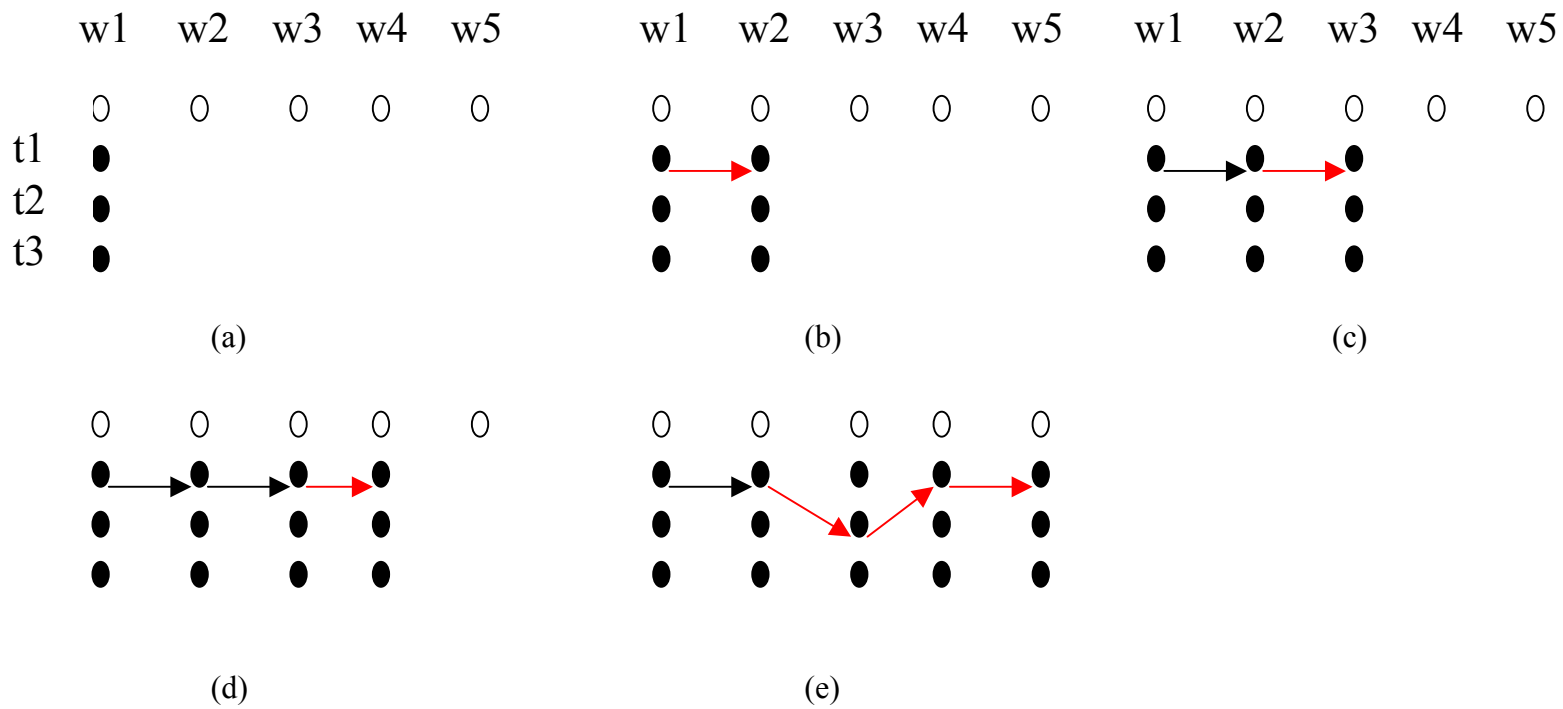
$$P(t_1,\ldots,t_n \mid w_1,\ldots,w_n)$$

$$= \prod_{i=1}^{n} p(t_i \mid w_1^n, t_1^{i-1})$$

$$\approx \prod_{i=1}^{n} p(t_i \mid h_i) = \prod_{i=1}^{n} Z(x_i)\exp(\sum_j \lambda_j f_j(h_i,t_i))$$

An example of search algorithm

- A search procedure with #words=5,K=3



Some applications

- Word Segmentation
- POS tagging
- ✓**Named entity recognition**
- WSD

Named Entity Recognition

(from **Hai Leong Chieu & Hwee Tou Ng**)

- Modeled as a classification problem
- Each token is assigned one of 29 (= 7*4 + 1) classes: 7 types of named entities(person, place, organization,… see MUC)
  - person_begin, person_continue, person_end, person_unique
  - org_begin, org_continue, org_end, org_unique,
  - …
  - nn (not-a-name)

An example

Consuela Washington , a longtime

person_begin person_end nn nn nn

House staffer ... the Securities and

org_unique nn nn org_begin org_continue

Exchange Commission in the Clinton ...

org_continue org_end nn nn person_unique

Local Features

- InitCapPeriod (e.g., *Mr.*)
- OneCap (e.g., *A*)
- AllCapsPeriod (e.g., *CORP.*)
- ContainDigit (e.g., *AB3*, *747*)
- TwoD (e.g., *99*)
- FourD (e.g., *1999*)
- DigitSlash (e.g., *01/01*)
- Dollar (e.g., *US$20*)
- Percent (e.g., *20%*)
- DigitPeriod (e.g., *$US3.20*)

Local Features

- Dictionary word lists
  - Person first names, person last names, organization names, location names
  - Downloaded from Internet
- Person prefix list (e.g., *Mr.*, *Dr.*), corporate suffix list (e.g., *Corp.*, *Inc.*)
  - Obtained from training data
- Month names, Days of the week, Numbers

Some applications

- Word Segmentation
- POS tagging
- Named entity recognition

✓**WSD**

Conclusion

- Any feature you think important can be used in the uniform model!
- It is a conditional model (different from generative model).
- It is successful in NLP

Project-I

- Multi_Word Organization Name Recognition
- Data(Training(70%) + Testing(30%)):
  http://www.icl.pku.edu.cn/icl_res/
  (人民日报切分/标注语料库下载)
- Requirement:
  - C/C++ Demo(including):
    - Training_tool
    - Test_tool (remove the tag of org_names first and then test!)
    - Evaluation_Tool
  - Experiment Report

What is multi_word Org_Name?

- Org_Tag: nt
- Single_word Org_Name:
  - Ex. 北京大学，中共中央
- Multi_Word Org_Name:
  - Ex. [北京市/ns 公安局/n]nt
  - [中国/ns 常驻/v 联合国/nt 日内瓦/ns 办事处/n]nt
- Only Multi_word is considered!

Group

- Each group contains 2 people.
- Use at least one approach

Reference

1. Dan Klein and Chirf Manning: Maxent Model, Conditional Estimation and Optimization. Tutorial of HLT-NAACL2003 and ACL2003
2. Stephen Della Pietra et al : Inducing Features of Random Fields. IEEE Trans. Pattern Analysis & Machine Intellegence, Vol. 19, No. 4, 1997
3. A maximum entropy approach to named entities recognition. Andrew Borthwick, PHD thesis.1999.
4. Maximum entropy model for natural language ambiguity resolution. Adwait Ratnaparkli, PHD thesis,1998

5. James R. Curran & Stephen Clark. Investigating GIS and Smoothing for Maximum Entropy Taggers. EACL2003.
6. Joshua Goodman. Exponential Priors for Maximum Entropy Models