6.891: Lecture 11 (October 15th, 2003)

Machine Translation Part II

Overview

- The Structure of IBM Models 1 and 2
- EM Training of Models 1 and 2
- Some examples of training Models 1 and 2
- IBM Model 3

Recap: IBM Model 1

- Aim is to model the distribution

$$P(\mathbf{f} \mid \mathbf{e})$$

where **e** is an English sentence $e_1 \ldots e_l$
f is a French sentence $f_1 \ldots f_m$

- Only parameters in Model 1 are **translation parameters**:

$$\mathbf{T}(f \mid e)$$

where f is a French word, e is an English word

- e.g.,

$$\begin{aligned} \mathbf{T}(le \mid the) &= 0.7 \\ \mathbf{T}(la \mid the) &= 0.2 \\ \mathbf{T}(l' \mid the) &= 0.1 \end{aligned}$$

Recap: Alignments in IBM Model 1

- Aim is to model the distribution

$$P(\mathbf{f} \mid \mathbf{e})$$

  where $\mathbf{e}$ is an English sentence $e_1 \ldots e_l$
  $\mathbf{f}$ is a French sentence $f_1 \ldots f_m$

- An **alignment** $\mathbf{a}$ identifies which English word each French word originated from

- Formally, an **alignment a** is $\{a_1, \ldots a_m\}$, where each $a_j \in \{0 \ldots l\}$.

- There are $(l+1)^m$ possible alignments.
  In IBM model 1 all alignments $\mathbf{a}$ are equally likely:

$$P(\mathbf{a} \mid \mathbf{e}) = C \times \frac{1}{(l+1)^m}$$

  where $C = prob(length(\mathbf{f}) = m)$ is a constant.

IBM Model 1: The Generative Process

To generate a French string f from an English string e:

- **Step 1:** Pick the length of f (all lengths equally probable, probability C)
- **Step 2:** Pick an alignment $\mathbf{a}$ with probability $\frac{1}{(l+1)^m}$
- **Step 3:** Pick the French words with probability

$$P(\mathbf{f} \mid \mathbf{a}, \mathbf{e}) = \prod_{j=1}^{m} \mathbf{T}(f_j \mid e_{a_j})$$

The final result:

$$P(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = P(\mathbf{a} \mid e) P(\mathbf{f} \mid \mathbf{a}, e) = \frac{C}{(l+1)^m} \prod_{j=1}^{m} \mathbf{T}(f_j \mid e_{a_j})$$

IBM Model 2

- Only difference: we now introduce **alignment** or **distortion** parameters

$$\mathbf{D}(i \mid j, l, m) \quad = \quad \text{Probability that } j\text{'th French word is connected to } i\text{'th English word, given sentence lengths of } \mathbf{e} \text{ and } \mathbf{f} \text{ are } l \text{ and } m \text{ respectively}$$

- Define

$$P(\mathbf{a} = \{a_1, \ldots a_m\} \mid \mathbf{e}, l, m) = \prod_{j=1}^{m} \mathbf{D}(a_j \mid j, l, m)$$

- Gives

$$P(\mathbf{f}, \mathbf{a} \mid \mathbf{e}, l, m) = \prod_{j=1}^{m} \mathbf{D}(a_j \mid j, l, m)\mathbf{T}(f_j \mid e_{a_j})$$

- Note: Model 1 is a special case of Model 2, where $\mathbf{D}(i \mid j, l, m) = \frac{1}{l+1}$ for all i, j.

An Example

$$
\begin{aligned}
l &= 6 \\
m &= 7 \\
\mathbf{e} &= \text{And the program has been implemented} \\
\mathbf{f} &= \text{Le programme a ete mis en application} \\
\mathbf{a} &= \{2, 3, 4, 5, 6, 6, 6\}
\end{aligned}
$$

$$
\begin{aligned}
P(\mathbf{a} \mid \mathbf{e}, l = 6, m = 7) = \; & \mathbf{D}(i = 2 \mid j = 1, l = 6, m = 7) \times \\
& \mathbf{D}(i = 3 \mid j = 2, l = 6, m = 7) \times \\
& \mathbf{D}(i = 4 \mid j = 3, l = 6, m = 7) \times \\
& \mathbf{D}(i = 5 \mid j = 4, l = 6, m = 7) \times \\
& \mathbf{D}(i = 6 \mid j = 5, l = 6, m = 7) \times \\
& \mathbf{D}(i = 6 \mid j = 6, l = 6, m = 7) \times \\
& \mathbf{D}(i = 6 \mid j = 7, l = 6, m = 7)
\end{aligned}
$$

$$
\begin{aligned}
P(\mathbf{f} \mid \mathbf{a}, \mathbf{e}) \quad = \quad & \mathbf{T}(Le \mid the) \times \\
& \mathbf{T}(programme \mid program) \times \\
& \mathbf{T}(a \mid has) \times \\
& \mathbf{T}(ete \mid been) \times \\
& \mathbf{T}(mis \mid implemented) \times \\
& \mathbf{T}(en \mid implemented) \times \\
& \mathbf{T}(application \mid implemented)
\end{aligned}
$$

IBM Model 2: The Generative Process

To generate a French string $\mathbf{f}$ from an English string $\mathbf{e}$:

- **Step 1:** Pick the length of **f** (all lengths equally probable, probability C)

- **Step 2:** Pick an alignment $\mathbf{a} = \{a_1, a_2 \ldots a_m\}$ with probability

$$\prod_{j=1}^{m} \mathbf{D}(a_j \mid j, l, m)$$

- **Step 3:** Pick the French words with probability

$$P(\mathbf{f} \mid \mathbf{a}, \mathbf{e}) = \prod_{j=1}^{m} \mathbf{T}(f_j \mid e_{a_j})$$

The final result:

$$P(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = P(\mathbf{a} \mid \mathbf{e}) P(\mathbf{f} \mid \mathbf{a}, \mathbf{e}) = C \prod_{j=1}^{m} \mathbf{D}(a_j \mid j, l, m) \mathbf{T}(f_j \mid e_{a_j})$$

Overview

- The Structure of IBM Models 1 and 2
- EM Training of Models 1 and 2
- Some examples of training Models 1 and 2
- IBM Model 3

A Hidden Variable Problem

- **We have:**

$$P(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) = C \prod_{j=1}^{m} \mathbf{D}(a_j \mid j, l, m)\mathbf{T}(f_j \mid e_{a_j})$$

- **And:**

$$P(\mathbf{f} \mid \mathbf{e}) = \sum_{\mathbf{a} \in \mathcal{A}} C \prod_{j=1}^{m} \mathbf{D}(a_j \mid j, l, m)\mathbf{T}(f_j \mid e_{a_j})$$

where $\mathcal{A}$ is the set of all possible alignments.

A Hidden Variable Problem

- Training data is a set of $(\mathbf{f}_k, \mathbf{e}_k)$ pairs, likelihood is

$$\sum_k \log P(\mathbf{f}_k \mid \mathbf{e}_k) = \sum_k \log \sum_{\mathbf{a} \in \mathcal{A}} P(\mathbf{a} \mid \mathbf{e}_k) P(\mathbf{f}_k \mid \mathbf{a}, \mathbf{e}_k)$$

  where $\mathcal{A}$ is the set of all possible alignments.

- We need to maximize this function w.r.t. the translation parameters, and the alignment probabilities

- EM can be used for this problem: initialize parameters randomly, and at each iteration choose

$$\Theta_t = \text{argmax}_\Theta \sum_i \sum_{\mathbf{a} \in \mathcal{A}} P(\mathbf{a} \mid \mathbf{e}_k, \mathbf{f}_k, \Theta^{t-1}) \log P(\mathbf{f}_k, \mathbf{a} \mid \mathbf{e}_k, \Theta)$$

  where Θ^t are the parameter values at the t'th iteration.

Models 1 and 2 Have a Simple Structure

- We have $\mathbf{f} = \{f_1 \dots f_m\}$, $\mathbf{a} = \{a_1 \dots a_m\}$, and

$$P(\mathbf{f}, \mathbf{a} \mid \mathbf{e}, l, m) = \prod_{j=1}^{m} P(a_j, f_j \mid \mathbf{e}, l, m)$$

  where

$$P(a_j, f_j \mid \mathbf{e}, l, m) = \mathbf{D}(a_j \mid j, l, m)\mathbf{T}(f_j \mid e_{a_j})$$

- **We can think of the m (f_j, a_j) pairs as being generated independently**

A Crucial Step in the EM Algorithm

- Say we have the following $(\mathbf{e}, \mathbf{f})$ pair:

$$\mathbf{e} = \text{And the program has been implemented}$$

$$\mathbf{f} = \text{Le programme a ete mis en application}$$

- Given that $\mathbf{f}$ was generated according to Model 2, what is the probability that $a_1 = 2$? **Formally:**

$$Prob(a_1 = 2 \mid \mathbf{f}, \mathbf{e}) = \sum_{\mathbf{a}:a_1=2} P(\mathbf{a} \mid \mathbf{f}, \mathbf{e}, l, m)$$

The Answer

$$
\begin{aligned}
Prob(a_1 = 2 \mid \mathbf{f}, \mathbf{e}) &= \sum_{\mathbf{a}:a_1=2} P(\mathbf{a} \mid \mathbf{f}, \mathbf{e}, l, m) \\
&= \frac{\mathbf{D}(a_1 = 2 \mid j = 1, l = 6, m = 7)\mathbf{T}(le \mid the)}{\sum_{i=0}^{l} \mathbf{D}(a_1 = i \mid j = 1, l = 6, m = 7)\mathbf{T}(le \mid e_i)}
\end{aligned}
$$

Follows directly because the (a_j, f_j) pairs are independent:

$$
\begin{aligned}
P(a_1 = 2 \mid \mathbf{f}, \mathbf{e}, l, m) &= \frac{P(a_1 = 2, f_1 = Le \mid f_2 \ldots f_m, \mathbf{e}, l, m)}{P(f_1 = Le \mid f_2 \ldots f_m, \mathbf{e}, l, m)} && (1) \\
&= \frac{P(a_1 = 2, f_1 = Le \mid \mathbf{e}, l, m)}{P(f_1 = Le \mid \mathbf{e}, l, m)} && (2) \\
&= \frac{P(a_1 = 2, f_1 = Le \mid \mathbf{e}, l, m)}{\sum_i P(a_1 = i, f_1 = Le \mid \mathbf{e}, l, m)}
\end{aligned}
$$

where (2) follows from (1) because $P(\mathbf{f}, \mathbf{a} \mid \mathbf{e}, l, m) = \prod_{j=1}^{m} P(a_j, f_j \mid \mathbf{e}, l, m)$

A General Result

$$
\begin{aligned}
Prob(a_j = i \mid \mathbf{f}, \mathbf{e}) &= \sum_{\mathbf{a}: a_j = i} P(\mathbf{a} \mid \mathbf{f}, \mathbf{e}, l, m) \\
&= \frac{\mathbf{D}(a_j = i \mid j, l, m)\mathbf{T}(f_j \mid e_i)}{\sum_{i'=0}^{l} \mathbf{D}(a_j = i' \mid j, l, m)\mathbf{T}(f_j \mid e_{i'})}
\end{aligned}
$$

Alignment Probabilities have a Simple Solution!

- e.g., Say we have $l = 6$, $m = 7$,

$$\mathbf{e} = \text{And the program has been implemented}$$

$$\mathbf{f} = \text{Le programme a ete mis en application}$$

- Probability of "mis" being connected to "the":

$$P(a_5 = 2 \mid \mathbf{f}, \mathbf{e}) = \frac{\mathbf{D}(a_5 = 2 \mid j = 5, l = 6, m = 7)\mathbf{T}(mis \mid the)}{Z}$$

where

$$\begin{aligned} Z = & \quad \mathbf{D}(a_5 = 0 \mid j = 5, l = 6, m = 7)\mathbf{T}(mis \mid NULL) \\ + & \quad \mathbf{D}(a_5 = 1 \mid j = 5, l = 6, m = 7)\mathbf{T}(mis \mid And) \\ + & \quad \mathbf{D}(a_5 = 2 \mid j = 5, l = 6, m = 7)\mathbf{T}(mis \mid the) \\ + & \quad \mathbf{D}(a_5 = 3 \mid j = 5, l = 6, m = 7)\mathbf{T}(mis \mid program) \\ + & \quad \ldots \end{aligned}$$

The EM Algorithm for Model 2

- Define

$$
\begin{array}{rl}
\mathbf{e}[k] & \text{for } k = 1 \ldots n \text{ is the } k\text{'th English sentence} \\
\mathbf{f}[k] & \text{for } k = 1 \ldots n \text{ is the } k\text{'th French sentence} \\
l[k] & \text{is the length of } \mathbf{e}[k] \\
m[k] & \text{is the length of } \mathbf{f}[k] \\
 & \\
\mathbf{e}[k,i] & \text{is the } i\text{'th word in } \mathbf{e}[k] \\
\mathbf{f}[k,j] & \text{is the } j\text{'th word in } \mathbf{f}[k]
\end{array}
$$

- Current parameters Θ^{t-1} are

$$
\begin{array}{rl}
\mathbf{T}(f \mid e) & \quad \text{for all } f \in \mathcal{F}, e \in \mathcal{E} \\
\mathbf{D}(i \mid j, l, m) &
\end{array}
$$

- We'll see how the EM algorithm re-estimates the $\mathbf{T}$ and $\mathbf{D}$ parameters

Step 1: Calculate the Alignment Probabilities

- Calculate an array of alignment probabilities (for $(k = 1 \ldots n)$, $(j = 1 \ldots m[k])$, $(i = 0 \ldots l[k])$):

$$
\begin{aligned}
a[i,j,k] &= P(a_j = i \mid \mathbf{e}[k], \mathbf{f}[k], \Theta^{t-1}) \\
&= \frac{\mathbf{D}(a_j = i \mid j, l, m)\mathbf{T}(f_j \mid e_i)}{\sum_{i'=0}^{l} \mathbf{D}(a_j = i' \mid j, l, m)\mathbf{T}(f_j \mid e_{i'})}
\end{aligned}
$$

where $e_i = \mathbf{e}[k,i]$, $f_j = \mathbf{f}[k,j]$, and $l = l[k], m = m[k]$

i.e., the probability of $\mathbf{f}[k,j]$ being aligned to $\mathbf{e}[k,i]$.

Step 2: Calculating the Expected Counts

- Calculate the translation counts

$$tcount(e, f) = \sum_{\substack{i,j,k: \\ \mathbf{e}[k,i]=e, \\ \mathbf{f}[k,j]=f}} a[i, j, k]$$

- $tcount(e, f)$ is expected number of times that e is aligned with f in the corpus

Step 2: Calculating the Expected Counts

- Calculate the source counts

$$scount(e) = \sum_{\substack{i,k: \\ \mathbf{e}[k,i]=e}} \sum_{j=1}^{m[k]} a[i,j,k]$$

- $scount(e)$ is expected number of times that e is aligned with any French word in the corpus

Step 2: Calculating the Expected Counts

- Calculate the alignment counts

$$acount(i, j, l, m) = \sum_{\substack{k: \\ l[k]=l, m[k]=m}} a[i, j, k]$$

$$acount(j, l, m) = |\{k : l[k] = l, m[k] = m\}|$$

- Here, $acount(i, j, l, m)$ is expected number of times that e_i is aligned to f_j in English/French sentences of lengths l and m respectively

- $acount(j, l, m)$ is number of times that we have sentences $\mathbf{e}$ and $\mathbf{f}$ of lengths l and m respectively

Step 3: Re-estimating the Parameters

- New translation probabilities are then defined as

$$P(f \mid e) = \frac{tcount(e, f)}{scount(e)}$$

- New alignment probabilities are defined as

$$P(a_j = i \mid j, l, m) = \frac{acount(i, j, l, m)}{acount(j, l, m)}$$

This defines the mapping from Θ^{t-1} to Θ^{t}

A Summary of the EM Procedure

- Start with parameters Θ^{t-1} as

$$\begin{array}{rl} \mathbf{T}(f \mid e) & \text{for all } f \in \mathcal{F}, e \in \mathcal{E} \\ \mathbf{D}(i \mid j, l, m) & \end{array}$$

- Calculate **alignment probabilities** under current parameters

$$a[i, j, k] = \frac{\mathbf{D}(a_j = i \mid j, l, m)\mathbf{T}(f_j \mid e_i)}{\sum_{i'=0}^{l} \mathbf{D}(a_j = i' \mid j, l, m)\mathbf{T}(f_j \mid e_{i'})}$$

- Calculate **expected counts** $tcount(e, f)$, $scount(e)$, $acount(i, j, l, m)$, and $acount(j, l, m)$ from the alignment probabilities

- Re-estimate $\mathbf{T}(f \mid e)$ and $\mathbf{D}(i \mid j, l, m)$ from the expected counts

The Special Case of Model 1

- Start with parameters Θ^{t-1} as

$$\mathbf{T}(f \mid e) \qquad \text{for all } f \in \mathcal{F}, e \in \mathcal{E}$$

(no alignment parameters)

- Calculate **alignment probabilities** under current parameters

$$a[i,j,k] \quad = \quad \frac{\mathbf{T}(f_j \mid e_i)}{\sum_{i'=0}^{l} \mathbf{T}(f_j \mid e_{i'})}$$

(because $\mathbf{D}(a_j = i \mid j, l, m) = 1/(l+1)^m$ for all i, j, l, m).

- Calculate **expected counts** $tcount(e, f)$, $scount(e)$,

- Re-estimate $\mathbf{T}(f \mid e)$ from the expected counts

Overview

- The Structure of IBM Models 1 and 2
- EM Training of Models 1 and 2
- Some examples of training Models 1 and 2
- IBM Model 3

An Example of Training Models 1 and 2

Example will use following translations:

e[1]	=	the	dog
f[1]	=	le	chien
e[2]	=	the	cat
f[2]	=	le	chat
e[3]	=	the	bus
f[3]	=	l'	autobus

NB: I won't use a NULL word e_0

Initial (random) parameters:

e	f	$\mathbf{T}(f \mid e)$
the	le	0.23
the	chien	0.2
the	chat	0.11
the	l’	0.25
the	autobus	0.21
dog	le	0.2
dog	chien	0.16
dog	chat	0.33
dog	l’	0.12
dog	autobus	0.18
cat	le	0.26
cat	chien	0.28
cat	chat	0.19
cat	l’	0.24
cat	autobus	0.03
bus	le	0.22
bus	chien	0.05
bus	chat	0.26
bus	l’	0.19
bus	autobus	0.27

Alignment probabilities:

i	j	k	a(i,j,k)
1	1	0	0.526423237959726
2	1	0	0.473576762040274
1	2	0	0.552517995605817
2	2	0	0.447482004394183
1	1	1	0.466532602066533
2	1	1	0.533467397933467
1	2	1	0.356364544422507
2	2	1	0.643635455577493
1	1	2	0.571950438336247
2	1	2	0.428049561663753
1	2	2	0.439081311724508
2	2	2	0.560918688275492

Expected counts:

e	f	$tcount(e, f)$
the	le	0.99295584002626
the	chien	0.552517995605817
the	chat	0.356364544422507
the	l’	0.571950438336247
the	autobus	0.439081311724508
dog	le	0.473576762040274
dog	chien	0.447482004394183
dog	chat	0
dog	l’	0
dog	autobus	0
cat	le	0.533467397933467
cat	chien	0
cat	chat	0.643635455577493
cat	l’	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l’	0.428049561663753
bus	autobus	0.560918688275492

Old and new parameters:

e	f	old	new
the	le	0.23	0.34
the	chien	0.2	0.19
the	chat	0.11	0.12
the	l’	0.25	0.2
the	autobus	0.21	0.15
dog	le	0.2	0.51
dog	chien	0.16	0.49
dog	chat	0.33	0
dog	l’	0.12	0
dog	autobus	0.18	0
cat	le	0.26	0.45
cat	chien	0.28	0
cat	chat	0.19	0.55
cat	l’	0.24	0
cat	autobus	0.03	0
bus	le	0.22	0
bus	chien	0.05	0
bus	chat	0.26	0
bus	l’	0.19	0.43
bus	autobus	0.27	0.57

e	f						
the	le	0.23	0.34	0.46	0.56	0.64	0.71
the	chien	0.2	0.19	0.15	0.12	0.09	0.06
the	chat	0.11	0.12	0.1	0.08	0.06	0.04
the	l’	0.25	0.2	0.17	0.15	0.13	0.11
the	autobus	0.21	0.15	0.12	0.1	0.08	0.07
dog	le	0.2	0.51	0.46	0.39	0.33	0.28
dog	chien	0.16	0.49	0.54	0.61	0.67	0.72
dog	chat	0.33	0	0	0	0	0
dog	l’	0.12	0	0	0	0	0
dog	autobus	0.18	0	0	0	0	0
cat	le	0.26	0.45	0.41	0.36	0.3	0.26
cat	chien	0.28	0	0	0	0	0
cat	chat	0.19	0.55	0.59	0.64	0.7	0.74
cat	l’	0.24	0	0	0	0	0
cat	autobus	0.03	0	0	0	0	0
bus	le	0.22	0	0	0	0	0
bus	chien	0.05	0	0	0	0	0
bus	chat	0.26	0	0	0	0	0
bus	l’	0.19	0.43	0.47	0.47	0.47	0.48
bus	autobus	0.27	0.57	0.53	0.53	0.53	0.52

After 20 iterations:

e	f	
the	le	0.94
the	chien	0
the	chat	0
the	l’	0.03
the	autobus	0.02
dog	le	0.06
dog	chien	0.94
dog	chat	0
dog	l’	0
dog	autobus	0
cat	le	0.06
cat	chien	0
cat	chat	0.94
cat	l’	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l’	0.49
bus	autobus	0.51

Model 2 has several local maxima – good one:

e	f	$\mathbf{T}(f \mid e)$
the	le	0.67
the	chien	0
the	chat	0
the	l'	0.33
the	autobus	0
dog	le	0
dog	chien	1
dog	chat	0
dog	l'	0
dog	autobus	0
cat	le	0
cat	chien	0
cat	chat	1
cat	l'	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l'	0
bus	autobus	1

Model 2 has several local maxima – bad one:

e	f	$\mathbf{T}(f \mid e)$
the	le	0
the	chien	0.4
the	chat	0.3
the	l’	0
the	autobus	0.3
dog	le	0.5
dog	chien	0.5
dog	chat	0
dog	l’	0
dog	autobus	0
cat	le	0.5
cat	chien	0
cat	chat	0.5
cat	l’	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l’	0.5
bus	autobus	0.5

another bad one:

e	f	$\mathbf{T}(f \mid e)$
the	le	0
the	chien	0.33
the	chat	0.33
the	l’	0
the	autobus	0.33
dog	le	1
dog	chien	0
dog	chat	0
dog	l’	0
dog	autobus	0
cat	le	1
cat	chien	0
cat	chat	0
cat	l’	0
cat	autobus	0
bus	le	0
bus	chien	0
bus	chat	0
bus	l’	1
bus	autobus	0

- Alignment parameters for good solution:

$$\begin{aligned}
\mathbf{T}(i = 1 \mid j = 1, l = 2, m = 2) &= 1 \\
\mathbf{T}(i = 2 \mid j = 1, l = 2, m = 2) &= 0 \\
\mathbf{T}(i = 1 \mid j = 2, l = 2, m = 2) &= 0 \\
\mathbf{T}(i = 2 \mid j = 2, l = 2, m = 2) &= 1
\end{aligned}$$

log probability $= -1.91$

- Alignment parameters for first bad solution:

$$\begin{aligned}
\mathbf{T}(i = 1 \mid j = 1, l = 2, m = 2) &= 0 \\
\mathbf{T}(i = 2 \mid j = 1, l = 2, m = 2) &= 1 \\
\mathbf{T}(i = 1 \mid j = 2, l = 2, m = 2) &= 0 \\
\mathbf{T}(i = 2 \mid j = 2, l = 2, m = 2) &= 1
\end{aligned}$$

log probability $= -4.16$

- Alignment parameters for second bad solution:

$$\begin{aligned}
\mathbf{T}(i=1 \mid j=1, l=2, m=2) &= 0 \\
\mathbf{T}(i=2 \mid j=1, l=2, m=2) &= 1 \\
\mathbf{T}(i=1 \mid j=2, l=2, m=2) &= 1 \\
\mathbf{T}(i=2 \mid j=2, l=2, m=2) &= 0
\end{aligned}$$

log probability $= -3.30$

Improving the Convergence Properties of Model 2

- **Out of 100 random starts, only 60 converged to the best local maxima**

- Model 1 converges to the same, global maximum every time (see the Brown et. al paper)

- Method in IBM paper: run Model 1 to estimate $\mathbf{T}$ parameters, then use these as the initial parameters for Model 2

- In 100 tests using this method, Model 2 converged to the correct point every time.

Overview

- The Structure of IBM Models 1 and 2
- EM Training of Models 1 and 2
- Some examples of training Models 1 and 2
- IBM Model 3

IBM Model 3

- The plot thickens...
- A new type of parameter: **fertility parameters**
- A quite different structure to the model...

IBM Model 3: Step 1 in the Generative Process

- English sentence $\mathbf{e} = \{e_1 \ldots e_l\}$, want to model $P(\mathbf{f} \mid \mathbf{e})$

- **Step 1:** choose $l + 1$ **fertilities** $\{\phi_0 \ldots \phi_l\}$ with probability

$$P(\{\phi_0 \ldots \phi_l\} \mid \mathbf{e})$$

- ϕ_i is the number of French words that e_i will be aligned with

IBM Model 3: Fertility Parameters

- New type of parameter

$$\mathbf{F}(\phi \mid e) = \text{probability that } e \text{ is aligned with } \phi \text{ words}$$

- For example

$$
\begin{aligned}
\mathbf{F}(0 \mid the) &= 0.1 \\
\mathbf{F}(1 \mid the) &= 0.9 \\
\mathbf{F}(2 \mid the) &= 0 \\
&\dots \\
\mathbf{F}(0 \mid not) &= 0.01 \\
\mathbf{F}(1 \mid not) &= 0.09 \\
\mathbf{F}(2 \mid not) &= 0.9 \\
&\dots
\end{aligned}
$$

IBM Model 3: Fertility Parameters

- **Step 1:** choose $l + 1$ **fertilities** $\{\phi_0 \ldots \phi_l\}$ with probability

$$P(\{\phi_0 \ldots \phi_l\} \mid \mathbf{e}) = P(\phi_0 \mid \phi_1 \ldots \phi_l) \prod_{i=1}^{l} \mathbf{F}(\phi_i \mid e_i)$$

IBM Model 3: Fertility Parameters

- Modeling $P(\phi_0 \mid \phi_1 \ldots \phi_l)$

- Take a single parameter $p_1 \in [0 \ldots 1]$. Define

$$P(\phi_0 \mid \phi_1 \ldots \phi_l) = \frac{m!}{(m-\phi_0)!\phi_0!} p_1^{\phi_0} (1-p_1)^{m-\phi_0}$$

  where $m = \sum_{i=1}^{l} \phi_i$

- Probability of seeing ϕ_0 heads if you toss a coin with probability p_1 of heads m times

- Intuition: $m = \sum_{i=1}^{l} \phi_i$ words have been generated from "real" English words; for each of these m words we generate an additional word connected to $NULL$ with probability p_1

IBM Model 3: The Final Fertility Model

- **Step 1:** choose $l+1$ **fertilities** $\{\phi_0 \ldots \phi_l\}$ with probability

$$P(\{\phi_0 \ldots \phi_l\} \mid \mathbf{e}) = \frac{m!}{(m-\phi_0)!\phi_0!} p_1^{\phi_0}(1-p_1)^{m-\phi_0} \prod_{i=1}^{l} \mathbf{F}(\phi_i \mid e_i)$$

- Parameters of the model are

$$p_1$$

and

$$\mathbf{F}(\phi \mid e) \text{ for } \phi = \{0, 1, 2, \ldots\}, e \in \mathcal{E}$$

IBM Model 3: The Distortion and Translation Parameters

- **Step 2:** For each e_i, for $k = 1 \ldots \phi_i$, choose a position $\pi_{i,k} \in 1 \ldots m$ and a French word $f_{i,k}$ with probability

$$\prod_{i=0}^{l} \prod_{k=1}^{\phi_i} \mathbf{R}(\pi_{i,k} \mid i, l, m)\mathbf{T}(f_{i,k} \mid e_i)$$

- Note that we now have **reverse distortion parameters**

$$\mathbf{R}(j \mid i, l, m) = \text{probability of French position } j \text{ given it's generated from English position } i$$

- Before we had **distortion parameters**

$$\mathbf{D}(i \mid j, l, m) = \text{probability of English position } i \text{ given it's generating French position } j$$

IBM Model 3: Final Model

$$
\begin{aligned}
\phi &= \{\phi_0 \ldots \phi_m\} \\
\pi &= \{\pi_{i,k} : i = 0 \ldots m, k = 1 \ldots \phi_i\} \\
\mathbf{f2} &= \{f_{i,k} : i = 0 \ldots m, k = 1 \ldots \phi_i\}
\end{aligned}
$$

$$P(\phi, \pi, \mathbf{f2} \mid \mathbf{e}) =$$

$$\frac{m!}{(m-\phi_0)!\phi_0!} p_1^{\phi_0} (1-p_1)^{m-\phi_0} \left(\prod_{i=1}^{l} \mathbf{F}(\phi_i \mid e_i) \right) \left(\prod_{i=0}^{l} \prod_{k=1}^{\phi_i} \mathbf{R}(\pi_{i,k} \mid i, l, m) \mathbf{T}(f_{i,k} \mid e_i) \right)$$

IBM Model 3: Alignments

- Note that given $\{\mathbf{f2}, \phi, \pi\}$, we can recover an alignment $\mathbf{a}$

- For a given alignment $\mathbf{a}$, we can calculate m, and ϕ, and there are

$$\prod_{i=0}^{l} \phi_i!$$

$\{\mathbf{f2}, \pi\}$ pairs which could produce the alignment from ϕ.

$$P(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) =$$

$$\frac{m!}{(m-\phi_0)!\phi_0!} p_1^{\phi_0} (1-p_1)^{m-\phi_0} \phi_0! \left(\prod_{i=1}^{l} \mathbf{F}(\phi_i \mid e_i) \phi_i! \right) \left(\prod_{i=0}^{l} \prod_{k=1}^{\phi_i} \mathbf{R}(\pi_{i,k} \mid i, l, m) \mathbf{T}(f_{i,k} \mid e_i) \right)$$

where $m, \phi, \pi,$ **f2** are direct functions of $\mathbf{a}$

A final simplification: $\mathbf{R}(\pi_{0,k} \mid i, l, m)$ is chosen so as to cancel the $\phi_0!$ term:

$$P(\mathbf{f}, \mathbf{a} \mid \mathbf{e}) =$$

$$\frac{m!}{(m-\phi_0)!\phi_0!} p_1^{\phi_0} (1-p_1)^{m-\phi_0} \left(\prod_{i=1}^{l} \mathbf{F}(\phi_i \mid e_i) \phi_i! \right) \left(\prod_{i=1}^{l} \prod_{k=1}^{\phi_i} \mathbf{R}(\pi_{i,k} \mid i, l, m) \right) \left(\prod_{i=0}^{l} \prod_{k=1}^{\phi_i} \mathbf{T}(f_{i,k} \mid e_i) \right)$$

where $m, \phi, \pi,$ **f2** are direct functions of $\mathbf{a}$

IBM Model 3: Summary

- Model 3 has the following parameter types

$$\begin{array}{rl} \mathbf{T}(f \mid e) & \text{translation parameters} \\ \mathbf{R}(j \mid i, l, m) & \text{(reverse) alignment parameters} \\ \mathbf{F}(\phi \mid e) & \text{fertility parameters} \\ p_1 & \text{parameter underlying } \phi_0 \end{array}$$

- Not possible to (efficiently) compute exact EM updates: we'll discuss approximations next class

- Note also that the model is **deficient**: assigns probability mass to "impossible" translations where different French words $f_{i,k}$ have the same position $\pi_{i,k}$