PCFG estimation with EM

The Inside-Outside Algorithm

Presentation order

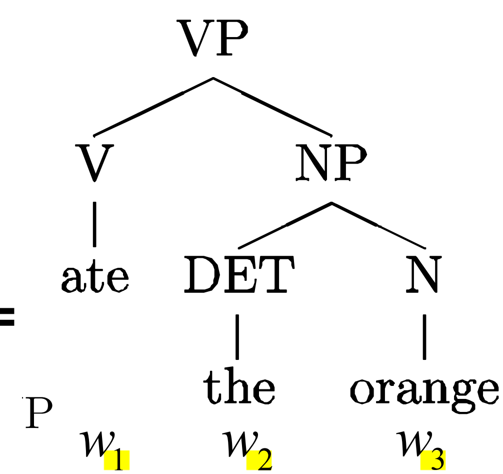
- Notation
- Calculating inside probabilities
- Calculating outside probabilities
- General schema for EM algorithms
- The inside-outside algorithm

Some notation

- $\{N^1, \ldots, N^n\}$
  Non-terminal symbols (hidden variables)
- $w_1 \ldots w_m = w_{1m}$
  Sentence (observed data)
- N^j_{pq}

  N^j spans $w_p \ldots w_q$ in string

$VP_{13} =$



Inside probability

- Definition:

$$\beta_j(p,q) = P(w_p \ldots w_q \mid N^j_{pq}, G) = P(N^j_{pq} \rightarrow w_{pq} \mid G)$$

- Computed recursively, base case:

$$\begin{aligned}\beta_j(k,k) &= P(w_k \mid N^j_{kk}, G) \\ &= P(N^j_{kk} \rightarrow w_k \mid G) \\ &= P(N^j \rightarrow w_k \mid G)\end{aligned}$$

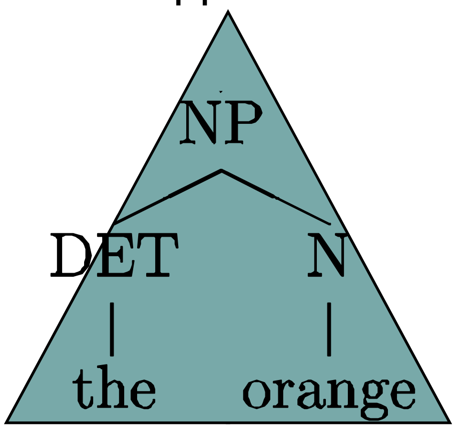
- Induction:

$$\beta_j(p,q) = \sum_{r,s} \sum_{d=p}^{q-1} P(N^j \rightarrow N^r N^s) \beta_r(p,d) \beta_s(d+1,q)$$

Inside probability example

- Consider the following PCFG fragment

NP→DET N	0.8	NP→N	0.2
DET→a	0.6	DET→the	0.4
N→apple	0.8	N→orange	0.2



$$\beta_{DET}(1,1) = P(the \mid DET_{11}, G) = P(DET \rightarrow the \mid G) = 0.4$$

$$\beta_{N}(2,2) = P(N \rightarrow orange \mid G) = 0.2$$

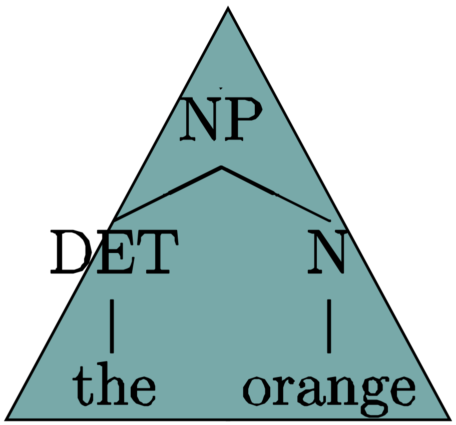
$$\begin{aligned}\beta_{NP}(1,2) &= P(NP \rightarrow DET \cdot N)\beta_{DET}(1,1)\beta_{N}(2,2) \\ &= 0.8 \qquad \times 0.4 \qquad \times 0.2\end{aligned}$$

$$\beta_{NP}(1,2) = 0.064$$

Inside probability example

- Consider the following PCFG fragment

NP→DET N	0.8	NP→N	0.2
DET→a	0.6	DET→the	0.4
N→apple	0.8	N→orange	0.2



$$\beta_{DET}(1,1) = P(the \mid DET_{11}, G) = P(DET \rightarrow the \mid G) = 0.4$$

$$\beta_{N}(2,2) = P(N \rightarrow orange \mid G) = 0.2$$

$$\beta_{NP}(1,2) = P(NP \rightarrow DET \cdot N)\beta_{DET}(1,1)\beta_{N}(2,2)$$
$$= 0.8 \qquad \times 0.4 \qquad \times 0.2$$

$$\beta_{NP}(1,2) = 0.064$$

- What is the probability of a sentence under a PCFG?

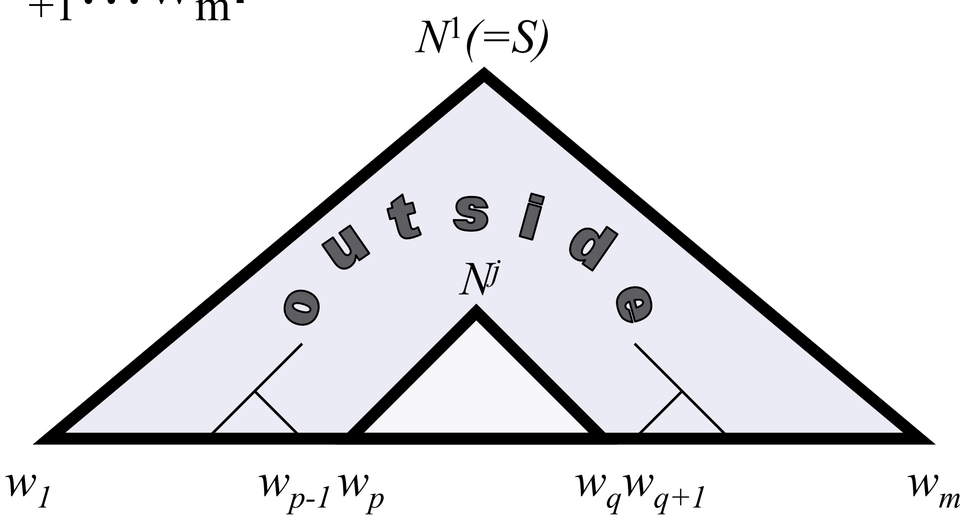
$$\beta_S(1,m) = P(S \rightarrow w_1 \ldots w_m \mid G)$$

Outside probability

- Definition

$$\alpha_j(p,q) = P(w_{1(p-1)}, N^j_{pq}, w_{(q+1)m} \mid G)$$

- That is, the joint probability of starting with N^1 (commonly called S) and generating words $w_1 \ldots w_{p-1}$,the non-terminal N^j, and words $w_{q+1} \ldots w_m$.



Calculating outside probability

- Computed recursively, base case:

$$\alpha_1(1,m) = 1 \qquad \alpha_S(1,m) = 1$$

$$\alpha_{j\neq 1}(1,m) = 0$$

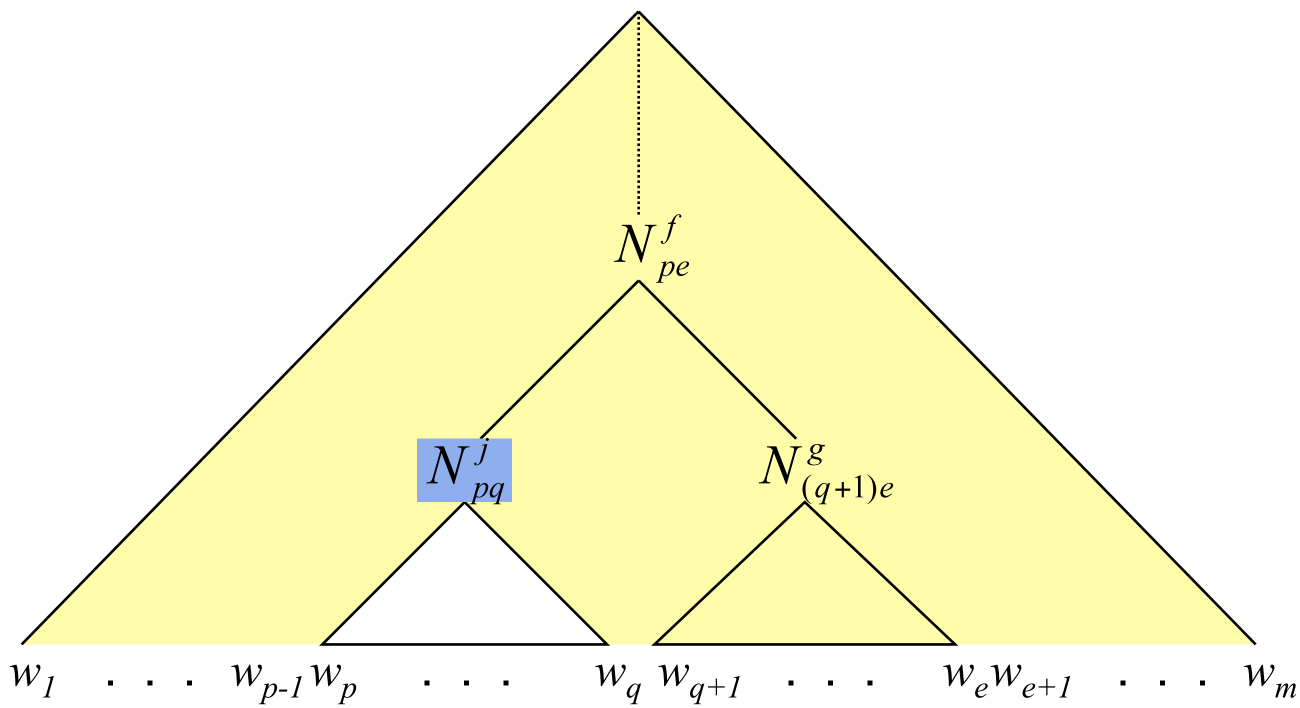
- How do we calculate $\alpha_j(p,q)$ in terms of this base case?
- Recall the definition:

$$\alpha_j(p,q) = P(w_{1(p-1)}, N^j_{pq}, w_{(q+1)m} \mid G)$$

- Intuition: N^j_{pq} must be either the L or R child of a parent node. We first consider the case when it is the L child.

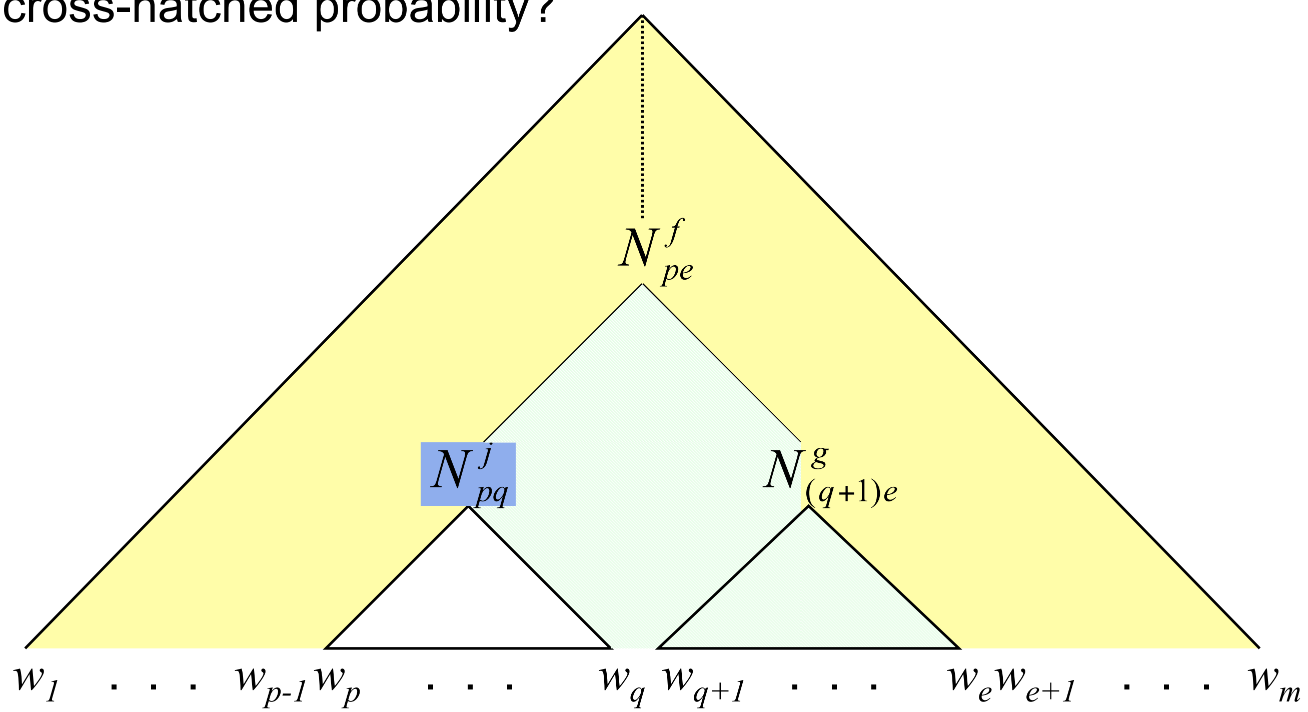
Outside probabilities: decomposing the problem

The shaded area represents the outside probability $\alpha_j(p,q)$ which we need to calculate. How can this be decomposed?



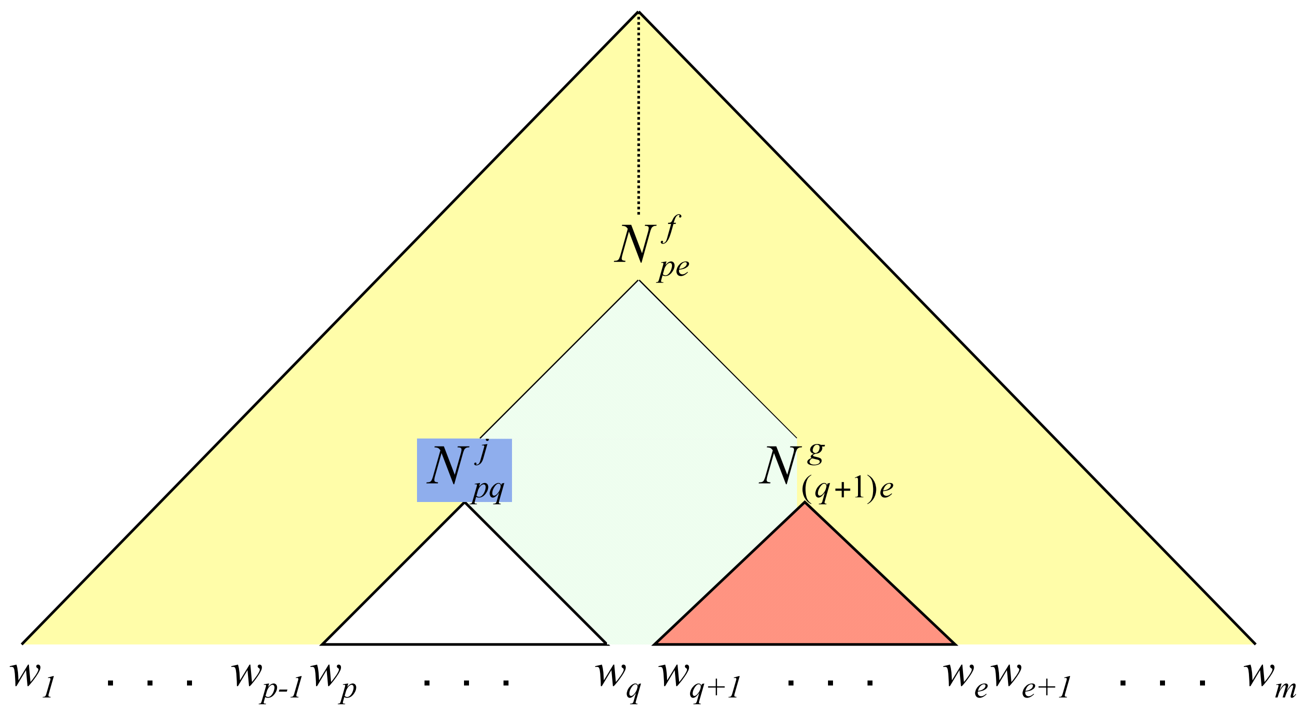
Outside probabilities: decomposing the problem

Step 1: We assume that N^f_{pe} is the parent of N^j_{pq}.
Its outside probability, $\alpha_f(p,e)$, (represented by the yellow shading) is available recursively. How do we calculate the cross-hatched probability?



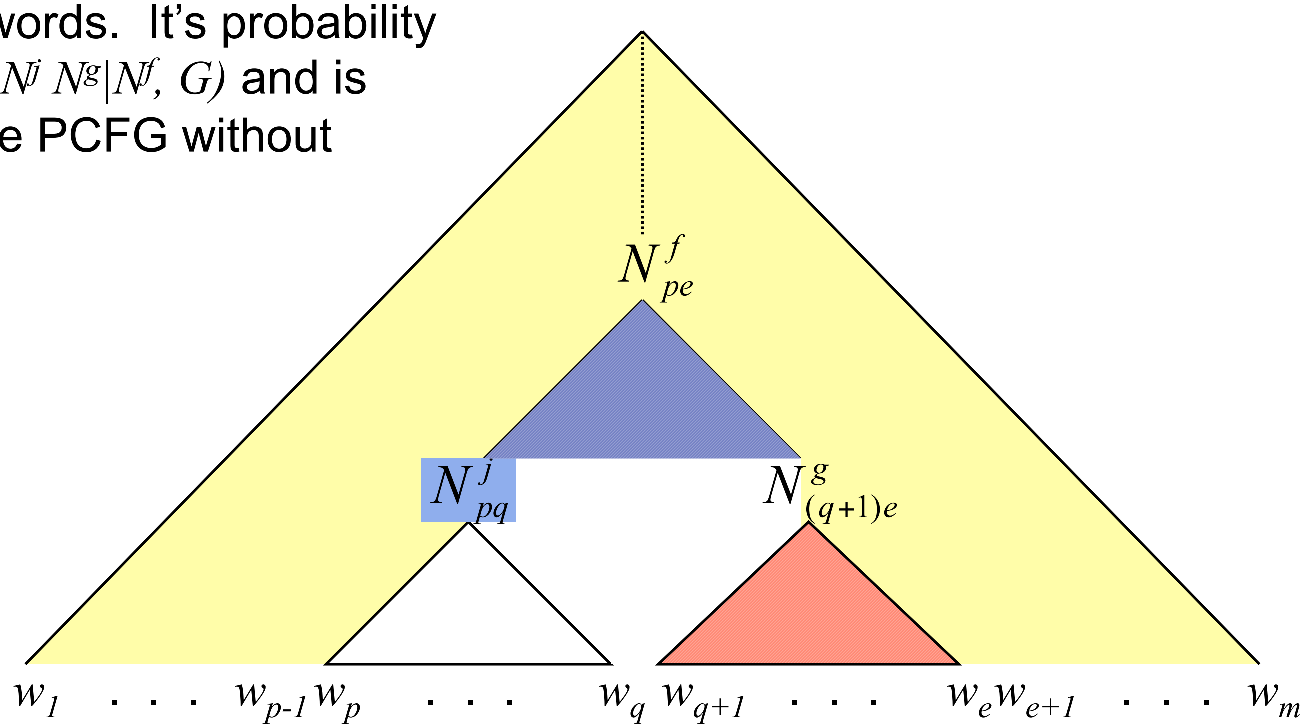
Outside probabilities: decomposing the problem

Step 2: The red shaded area is the inside probability of $N^g_{(q+1)e}$, which is available as $\beta_g(q+1,e)$.



Outside probabilities: decomposing the problem

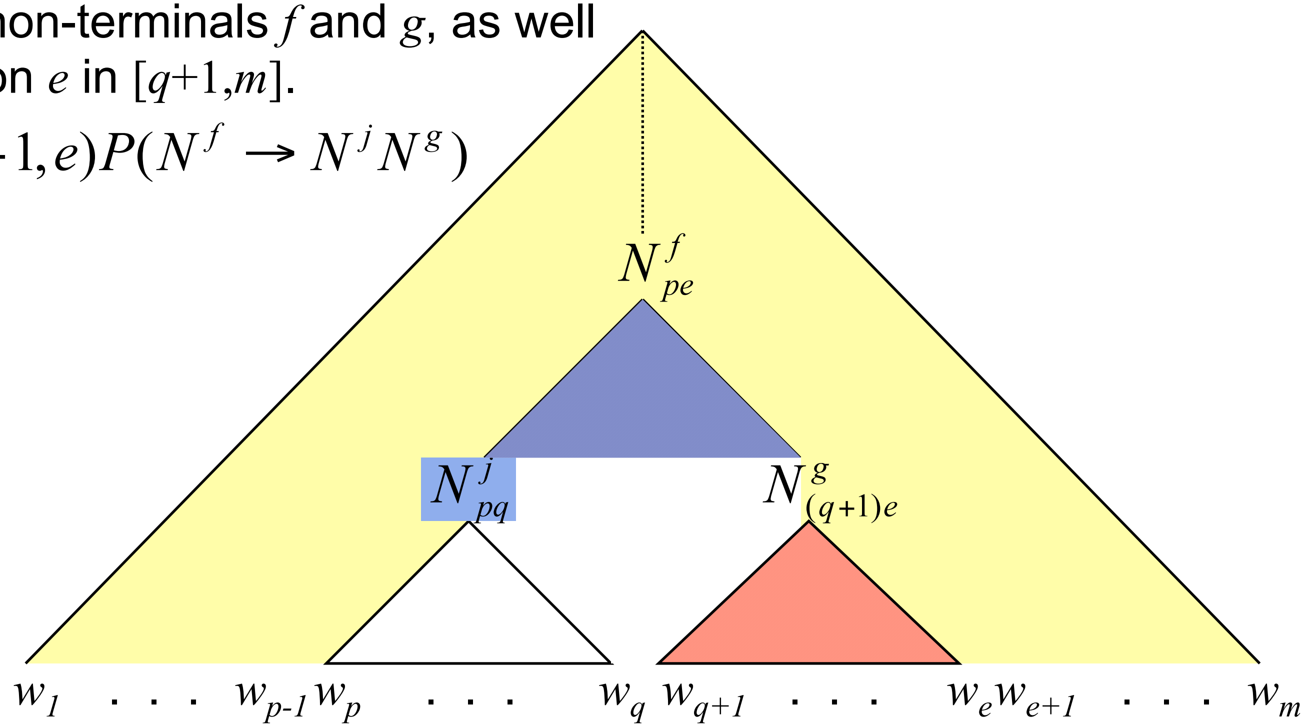
Step 3: The blue shaded part corresponds to the production $N^f \rightarrow N^j\ N^g$, which because of the context-freeness of the grammar, is not dependent on the positions of the words. It's probability is simply $P(N^f \rightarrow N^j\ N^g | N^f, G)$ and is available from the PCFG without calculation.



Outside probabilities: decomposing the problem

Multiplying the terms together, we have the joint probability corresponding to the yellow, red, and blue areas, assuming N^j was the left child of N^f, and given fixed non-terminals f and g, as well as a fixed partition e in $[q+1,m]$.

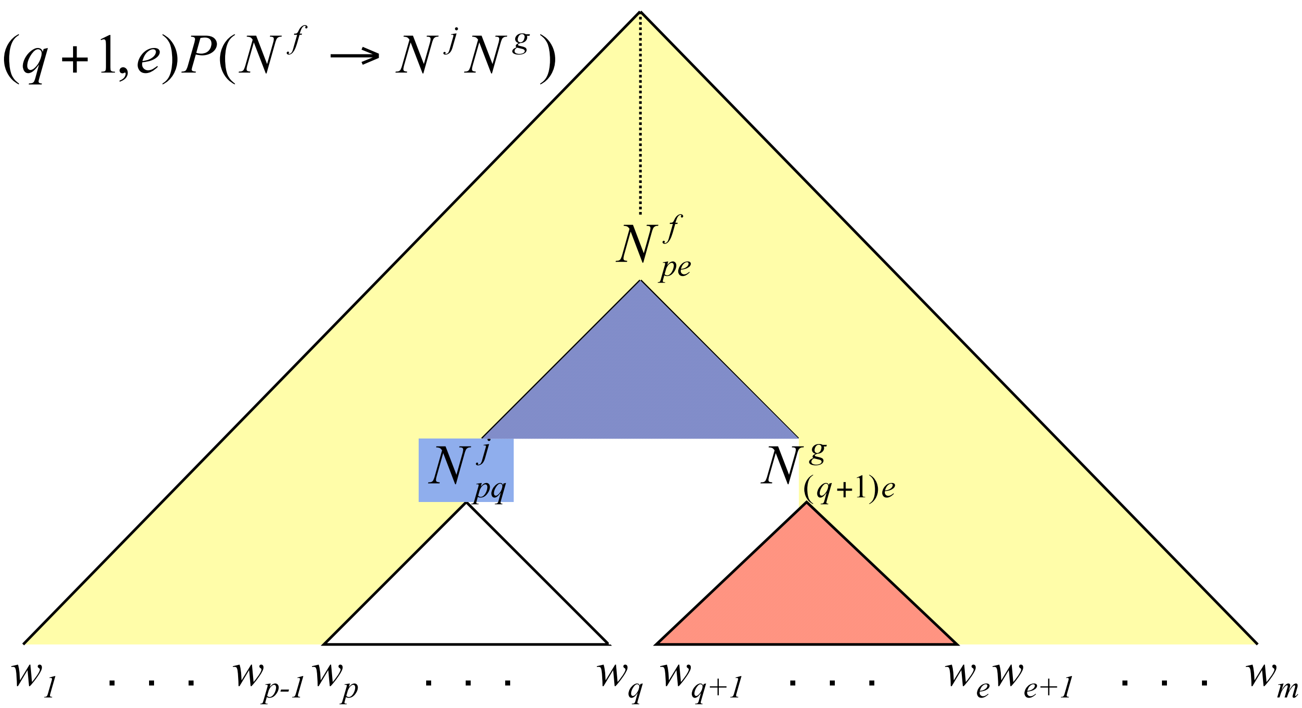
$$\alpha_f(p,q)\beta_g(q+1,e)P(N^f \rightarrow N^j N^g)$$



Outside probabilities: decomposing the problem

The total joint probability for a left sided N^j can be calculated by summing over all non-terminals f and g and partition e.

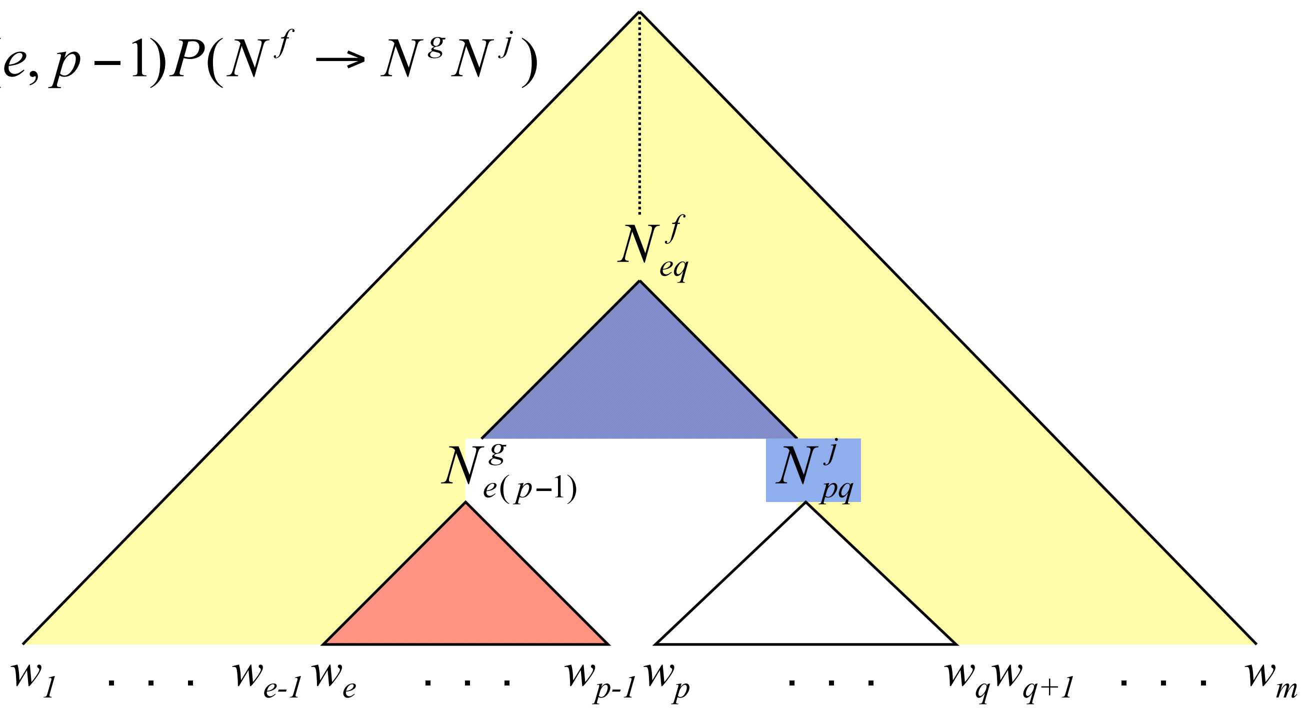
$$\sum_{f,g}\sum_{e=q+1}^{m}\alpha_f(p,q)\beta_g(q+1,e)P(N^f \rightarrow N^j N^g)$$



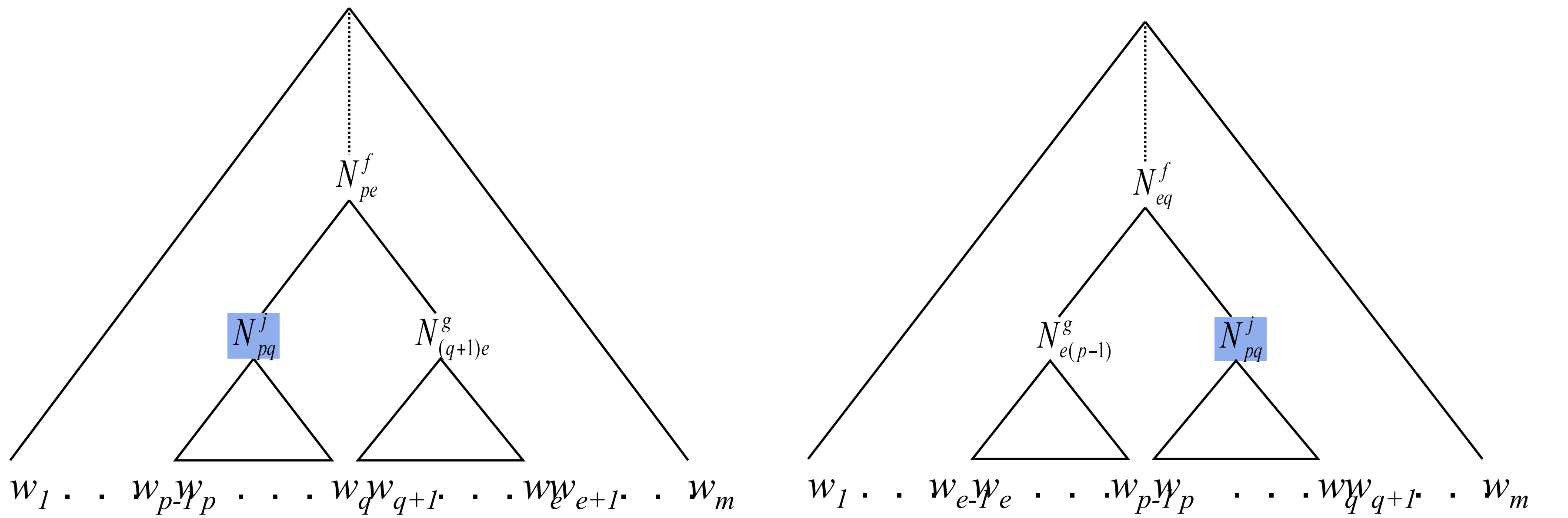
Outside probabilities: decomposing the problem

The total joint probability for a right-sided N^j is shown schematically in this diagram. The relevant calculation is:

$$\sum_{f,g}\sum_{e=1}^{p-1}\alpha_f(p,q)\beta_g(e,p-1)P(N^f \rightarrow N^g N^j)$$



Calculating the outside probability: final form



Since N^j may be either the left or right child, we have to add both terms. And, since $N^f \rightarrow N^j\, N^g / N^g\, N^j$ will get counted twice when $g=j$, it must be discounted on one side.

$$\alpha_j(p,q) =$$

$$\sum_{f,g} \sum_{e=q+1}^{m} \alpha_f(p,q)\beta_g(q+1,e)P(N^f \rightarrow N^j N^g) + \sum_{f,g \neq j} \sum_{e=1}^{p-1} \alpha_f(p,q)\beta_g(e,p-1)P(N^f \rightarrow N^g N^j)$$

General schema for certain EM algorithms

- Given two events, x and y, the maximum likelihood estimation (MLE) for their conditional probability is:

$$P(x \mid y) = \frac{count(x, y)}{count(x)}$$

- If they are observable, it's easy to see what to do: just count the events in a representative corpus and use the MLE or a smoothed distribution.

General schema for certain EM algorithms

- What these are hidden variables that cannot be observed directly?

  Use a model μ and iteratively improve the model based on a corpus of observable data (O) generated by the hidden variables:

$$P_{\hat{\mu}}(x \mid y) = \frac{E_{\mu}[count(x, y) \mid O]}{E_{\mu}[count(x) \mid O]}$$

- It is worth noting that if you know how to calculate the numerator, the denominator is trivially derivable.

General schema for certain EM algorithms

- By updating μ and iterating, the model converges to at least a local maximum.

- This can be proven, but I will not do it here.

The inside-outside algorithm

- Goal: estimate a model μ that is a PCFG (in Chomsky normal form) that characterizes a corpus of text.

- Required input:
  - Size of non-terminal vocabulary, n
  - At least one sentence to be modeled, O

The inside-outside algorithm

- Stated with the general schema described earlier, we seek to the MLE probabilities for productions in the grammar.

$$P(N^j \to N^r N^s \mid N^j) = \frac{count(N^j \to N^r N^s, N^j)}{count(N^j)}$$

- (Observe that this would be trivially easy to calculate this with a treebank, since the non-terminals are observable in a treebank)

The inside-outside algorithm

- Since the non-terminals are not visible, we can use EM to estimate the probabilities iteratively:

$$P_{\hat{\mu}}(N^j \rightarrow N^r N^s \mid N^j) = \frac{E_\mu[count(N^j \rightarrow N^r N^s, N^j) \mid O]}{E_\mu[count(N^j) \mid O]}$$

The inside-outside algorithm

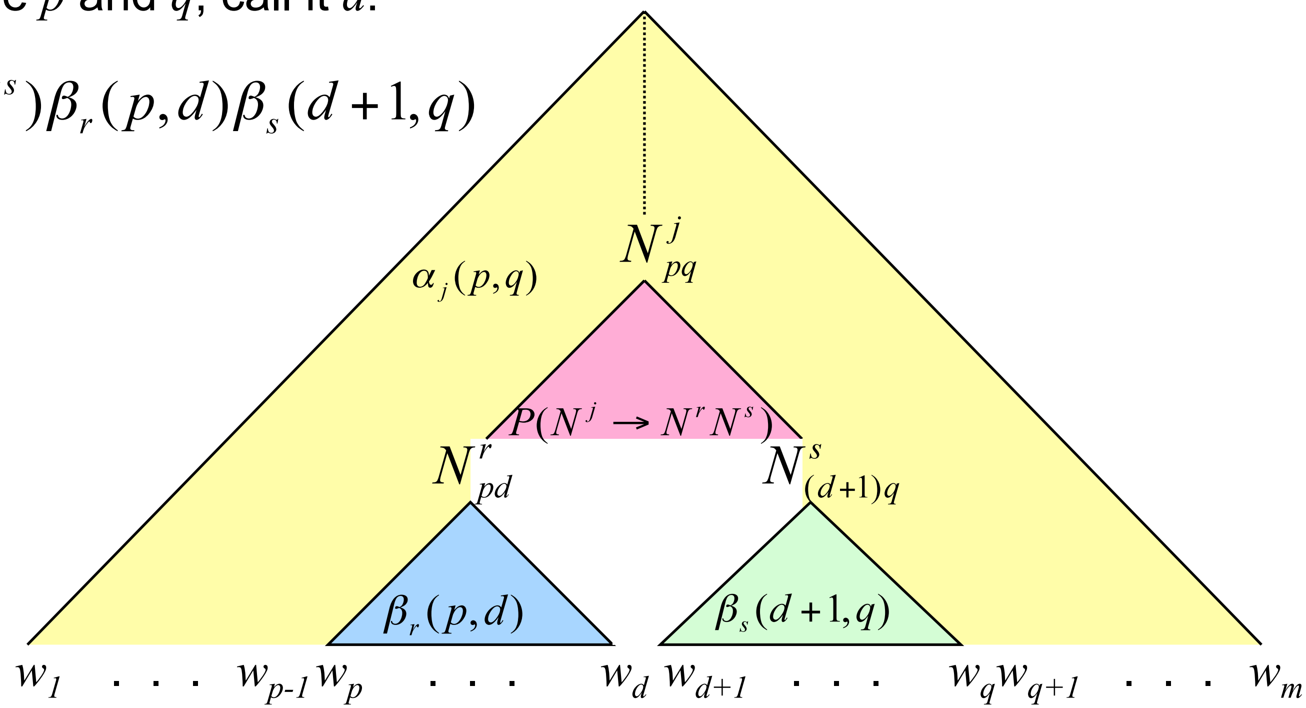
- We begin by taking the numerator alone:

$$E_\mu[count(N^j \rightarrow N^r N^s, N^j) \mid O]$$

The inside-outside algorithm

What we want is, for given non-terminals r and s, a probability that N^j is both used at some point in the derivation and accounts for span w_{pq}. Since there are two rules on the RHS, we need to pick a partition between the p and q, call it d:

$$\alpha_j(p,q)P(N^j \rightarrow N^r N^s)\beta_r(p,d)\beta_s(d+1,q)$$



The inside-outside algorithm

- Summing gives the total probability for any partition d:

$$= \alpha_j(p,q)P(N^j \rightarrow N^r N^s)\left[\sum_{d=p}^{q-1} \beta_r(p,d)\beta_s(d+1,q)\right]$$

- Expectation just involves summing the probabilities of all possible opportunities for using this rule in the derivation of w_{1m}. Each such opportunity is a span p,q of 2 words or more in w_{1m} (since we are dealing with binary rules).

$$E_\mu[count(N^j \rightarrow N^r N^s, N^j) \mid O] = \sum_{p=1}^{m} \sum_{p=q+1}^{m} P(N_{pq}^j \rightarrow N^r N^s \mid O, \mu)$$

The inside-outside algorithm

- We can use the definition of conditional probability to turn $P(N_{pq}^{j} \rightarrow N^{r}N^{s} \mid O,\mu)$ into $P(N_{pq}^{j} \rightarrow N^{r}N^{s},O,\mu)/P(O \mid \mu)$
- Therefore, the expected value of the numerator in the EM equation is

$$\sum_{p=1}^{m}\sum_{q=p+1}^{m}\frac{\alpha_{j}(p,q)P(N^{j} \rightarrow N^{r}N^{s})\left[\sum_{d=p}^{q-1}\beta_{r}(p,d)\beta_{s}(d+1,q)\right]}{P(O \mid \mu)}$$

- $P(O|\mu)$ is just the inside probability $\beta_{1}(1,m)$

The inside-outside algorithm

- Notice the analogy with the forward-backward algorithm.

Probability of getting from the start to the point where the latent event happens according to μ. (Outside ≈ Forward)

Probability of the latent event according to μ. (Rule ≈ Transition)

Probability of getting the rest of the way according to μ. (Inside ≈ Backward)

$$\underbrace{\sum_{p=1}^{m}\sum_{q=p+1}^{m}}\frac{\underbrace{\alpha_j(p,q)}\,\underbrace{P(N^j \rightarrow N^r N^s)}\,\underbrace{\left[\sum_{d=p}^{q-1}\beta_r(p,d)\beta_s(d+1,q)\right]}}{\underbrace{P(O \mid \mu)}}$$

Number of opportunities for the unobservable event to happen. (Spans ≈ Time steps)

Probability of the entire observed string being generated, according to μ (uses solution to "first fundamental problem")

The inside-outside algorithm

- What is the denominator $E_\mu[count(N^j) \mid O]$?

- One possibility is to calculate the value of the numerator and sum the result over all non-terminals r, s.

The inside-outside algorithm

- Also, intuitively, it can be thought of as a sum of the probabilities over ALL spans in the w_{1m} that N^j generated. The probability for a production N^j in a given span p,q is:

$$P(N_{pq}^{j} \mid N_{1m}^{1}, \mu) = \frac{P(N_{pq}^{j} \mid \mu)}{P(N_{1m}^{1} \mid \mu)} = \frac{\alpha_{j}(p,q)\beta_{j}(p,q)}{\beta_{1}(1,m)}$$

- Thus, the expectation count of using the production in a given sentence is:

$$\sum_{p=1}^{m}\sum_{q=p}^{m}\frac{\alpha_{j}(p,q)\beta_{j}(p,q)}{\beta_{1}(1,m)}$$

The inside-outside algorithm

- Putting the pieces together yields:

$$P_{\hat{\mu}}(N^j \rightarrow N^r N^s \mid N^j) = \frac{\sum_{p=1}^{m} \sum_{q=p+1}^{m} \alpha_j(p,q) P(N^j \rightarrow N^r N^s) \left[\sum_{d=p}^{q-1} \beta_r(p,d) \beta_s(d+1,q) \right]}{\sum_{p=1}^{m} \sum_{q=p}^{m} \alpha_j(p,q) \beta_j(p,q)}$$

- Notice that the indices on the summations are slightly different. This is because the numerator deals exclusively with binary rules, which must span at least two terminals!