Organizing Web Information (CS 728)
Computer Science and Engineering
Indian Institute of Technology Bombay

Midterm Exam
2014-02-21 Friday
17:30–19:30 SIC201

NAME ~~~~~~~~~~~~~~~~~~~~ ROLL ~~~~~~~~~~~~

This exam has 8 printed page/s. Write your name and roll number on **EVERY SIDE (and not just sheet)**, because we may take apart your answer book and/or xerox it for correction. Write your answer clearly within the spaces provided and on any last blank page. Do not write inside the rectangles to be used for grading. **If you need more space than is provided, you probably made a mistake in interpreting the question.** Start with rough work elsewhere, but you need not attach rough work. Use the marks alongside each question for time management. **Illogical or incoherent answers are worse than wrong answers or even *no* answer, and may fetch negative credit.** You may not use any computing or communication device during the exam. You may use textbooks, class notes written by you, approved material downloaded **prior to the exam** from the course Web page, course news group, or the Internet, or notes made available by me for xeroxing. If you use class notes from other student/s, you must obtain them **prior to the exam** and **write down his/her/their name/s and roll number/s** here.

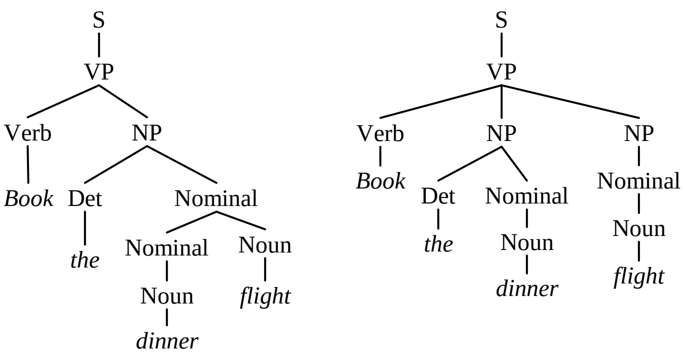
1. A context-free grammar (CFG) is a 4-tuple $G = (N, \Sigma, R, S)$ where:

- N is a finite set of non-terminal symbols.
- Σ is a finite set of terminal symbols.
- R is a finite set of rules of the form $X \to Y_1 Y_2 \ldots Y_n$, where $X \in N$, $n \geq 0$, and $Y_i \in N \cup \Sigma$ for $i = 1, \ldots, n$. For simplicity we will assume that $n = 1$ and $n = 2$ for productions to terminals and non-terminals respectively.
- $S \in N$ is a distinguished start symbol.

A PCFG, in addition, has a parameter $q(\alpha \to \beta) \geq 0$ for each rule $\alpha \to \beta \in R$, which can be interpreted as the conditional probability of choosing this rule in a left-most derivation, given that the non-terminal being expanded is α. For any $X \in N$, $\sum_\beta q(X \to \beta) = 1$.

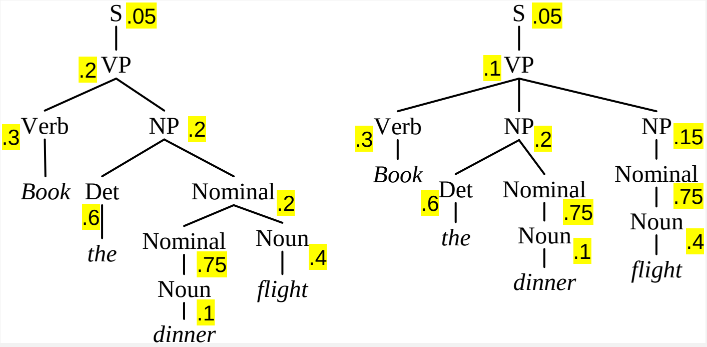
1(a) Given a production table with probabilities (there is some redundancy that you can ignore) and two possible parses of a sentence, write down the probability generating the sentence from the two parse trees. Mark relevant probabilities directly on the trees, and write the final answer alongside.

Rules	q
S → VP	.05
VP → Verb NP	.2
VP → Verb NP NP	.1
NP → Nominal	.15
NP → Det Nominal	.2
Nominal → Nominal Noun	.2
Nominal → Noun	.75
Verb → *book*	.3
Det → *the*	.6
Noun → *dinner*	.1
Noun → *flight*	.4



4

Each internal node is shown annotated with the production probability taken from the table. The probability of the left tree is $.05 \times .2 \times .3 \times .2 \times .6 \times .2 \times .75 \times .4 \times .1 = 2.16 \times 10^{-6}$. The probability of the right tree is $.05 \times .1 \times .3 \times .2 \times .15 \times .6 \times .75 \times .75 \times .1 \times .4 = 6.075 \times 10^{-7}$. Final numerical result is not required for full credit.



1(b) Let $\mathcal{T}_G(s)$ be the set of all possible parse trees of sentence s, and $t_1 t_2 \in \mathcal{T}_G(s)$ be the above trees. We just found $\Pr(t_1)$ and $\Pr(t_2)$. We are now interested in the partition function or normalizer $\Pr(s) = \sum_{t \in \mathcal{T}_G(s)} \Pr(t)$. Given sentence $s = x_1, \ldots, x_n$, define $\mathcal{T}(i,j,\alpha)$ for $1 \le i \le j \le n$ and non-terminal $\alpha \in N$ as the set of parse trees for the span $x_i \ldots x_j$ such that α is at the root. Define $\pi(i,j,\alpha) = \sum_{t \in \mathcal{T}(i,j,\alpha)} \Pr(t)$. (Sum over the empty set is zero.) What is $\Pr(s)$ using elements of $\pi(i,j,\alpha)$?

1

$\Pr(s) = \pi(1,n,S)$.

1(c) The base case is, for all $i = 1, \ldots, n$ and all $\alpha \in N$,

$$\pi(i, \underline{\quad}, \alpha) = \begin{cases} q(\alpha \to \underline{\qquad}) & \text{if } \alpha \to \text{(same as previous)} \in R, \\ \underline{\qquad\qquad} & \text{otherwise} \end{cases}$$

(Complete.)

2

$$\pi(i, \utilde{i}, \alpha) = \begin{cases} q(\alpha \to \utilde{x_i}) & \text{if } \alpha \to x_i \in R, \\ \utilde{0} & \text{otherwise} \end{cases}$$

1(d) The recursive step is, for $1 \le i < j \le n$, and for all $\alpha \in N$,

$$\pi(i, j, \alpha) = \sum_{\alpha \to \utilde{\qquad\qquad}, k \in \utilde{\quad}} q(\utilde{\qquad\qquad\qquad})$$

$$\pi(\utilde{\quad}, \utilde{\quad}, \utilde{\quad})\pi(\utilde{\qquad}, \utilde{\quad}, \utilde{\quad}).$$

Complete with explanation.

		3

$$\pi(i, j, \alpha) = \sum_{\alpha \to \utilde{\beta\, \gamma}, k \in [\utilde{i, j-1}]} q(\utilde{\alpha \to \beta\, \gamma})\pi(\utilde{i}, \utilde{k}, \utilde{\beta})\pi(\utilde{k+1}, \utilde{j}, \utilde{\gamma}).$$

2. Consider a standard linear chain CRF. We wish to train it discriminatively using a structured SVM on sequences $x = x_1, \dots, x_T$, $y = y_1, \dots, y_T$, with K possible states for each y_t. To do that we need a loss-augmented inference algorithm to maximize $w \cdot \phi(x, y) + \Delta(y^*, y)$. Suppose ϕ decomposes as $\phi(x, y) = \sum_{1 \le t \le T} \varphi(y_{t-1}, y_t, x_t)$, with a fixed sentinel y_0 (and no x_0).

2(a) First we consider the simple Hamming loss, defined as $\Delta_h(y^*, y) = \sum_t [\![y^*_t \ne y_t]\!]$. In this case, the loss-augmented inference objective is

$$\arg\max_y \sum_{t=1}^{T} w \cdot \varphi(y_{t-1}, y_t, x_t) + [\![y^*_t \ne y_t]\!]. \tag{1}$$

Write down

- the design (all indices, their value ranges, and meanings, the meaning of the contents of the table cells) of the dynamic programming table,
- the base case initializations,
- the recursive step for completing the loss augmented inference,
- and the time taken by your code for one instance.

		4

The table is $V(t,k)$ where $t = 0, 1, \ldots, T$ and $k = 1, \ldots, K$.

$$V(t,k) = \max_{\overrightarrow{y_1 \cdots y_{t-1} k}} \sum_{\tau=1}^{t} w \cdot \varphi(y_{\tau-1}, y_\tau, x_\tau) + [\![y^*_\tau \neq y_\tau]\!],$$

which is the incomplete sum up to t instead of T as in (1), and ending in state k. The base cases are $V(0,k) = 0$ for all state labels k. The inductive step is

$$V(t,k) = \max_{k'=1,\ldots,K} V(t-1,k') + w \cdot \varphi(k', k, x_t) + [\![y^*_t \neq k]\!].$$

The time to fill this table is $O(TK^2)$.

2(b) We describe a different loss function $\Delta(y^*, y)$ verbally. $\Delta(y^*, y) = \Gamma(\Delta_h(y^*, y))$, where $\Gamma(\bullet)$ is a function that expresses the trainer's dissatisfaction with the number of mistakes made by the learner. The case $\Gamma(\bullet) = \bullet$ takes us back to Δ_h, but, in general, we can have a "lenient grader" where Γ shows diminishing annoyance (e.g., $\Gamma(\bullet) = \sqrt{\bullet}$) with incorrect labels, or a "harsh grader", as in $\Gamma(\bullet) = \bullet^2$. Repeat the above steps for loss-augmented inference with this new loss function (assuming Γ as additional input), taking as little space and time as possible.

(Hint: it may help to start with a specific situation. For token positions that agree between y^* and y, there is no loss. For the first differing position, there is one unit of loss. For the second mistake, two units are assessed, for the third mistake, three units, and so on.)

		6

The difficulty in extending Hamming loss is that, at any t where $y^*_t \neq y_t$, the loss depends on the number of mistakes made before t. It is tempting but incorrect to try to do this using a second table $M(t,k)$, where $t = 0, \ldots, T$ and $K = 1, \ldots, K$. One correct solution is to extend the earlier dynamic programming table $V(t,k)$ to a third dimention and call it $V(t,m,k)$, the maximum objective value up to t positions, ending in state k, and having made $m \leq t$ mistakes. I.e., the new table has $O(T^2K)$ cells, in place of the $O(TK)$ cells before. Given a table with all cells filled with values, loss-augmented inference amounts to finding $\max_{m,k} V(T,m,k)$ (and tracing back the solution path). The base cases for filling $V(t,m,k)$ are:

- $V(0,0,k) = 0$ for all k.
- $V(t,m,k) = -\infty$ if $t < m$, for all k.

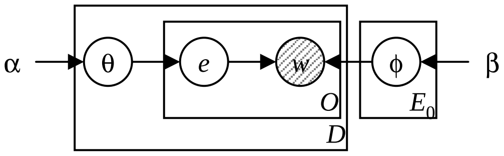
For $t > 0$ and $t \geq m$, cell $V(t,m,k)$ is filled as follows:

```
$best \leftarrow -\infty$
if $k = y^*_t$ then
    for $k' = 1, \ldots, K$ do
        $best \leftarrow \max\{best, V(t-1,m,k') + w \cdot \varphi(k', k, x_t)\}$
else$[k \neq y^*_t]$
    for $k' = 1, \ldots, K$ do
```

```
        $best \leftarrow \max\{best, V(t-1, m-1, k') + w\cdot\varphi(k', k, x_t) + \Gamma(m) - \Gamma(m-1)\}$
    $V(t, m, k) \leftarrow best$
The time taken to fill this table is $O(T^2K^2)$.
```

3. We will put together a LDA-like topic model and a Markovian token sequence model to address entity annotation in text.

3(a) First consider the direct adaptation of LDA with one topic representing each entity. Let all documents d in question be the same length and have O token offsets, $1, \ldots, o, \ldots, O$. Each d, o has a hidden entity variable e_{do}. The observed word is w_{do}. In the plate diagram below, α, β are scalar concentration parameters in Dirichlet priors (i.e., $\mathrm{Dir}(\alpha, \ldots, \alpha)$ etc.). ϕ_e is a per-entity multinomial distribution over words. θ_d is a multinomial distribution over entities to be mentioned in one document. E_0 is the universe of entities.



Given fixed α, β and observing w_{do}, the goal is to estimate posterior distributions for ϕ globally, and θ and e locally. Fixing a d and o, define counts

$$\begin{aligned}
n_e &= \sum_{d'} \sum_{o'} [\![e_{d'o'} = e]\!] \\
n_e^{\backslash d,o} &= \sum_{d'} \sum_{o'} [\![e_{d'o'} = e]\!][\![d \neq d' \vee o \neq o']\!] \\
n_{w,e} &= \sum_{d'} \sum_{o'} [\![e_{d'o'} = e]\!][\![w_{d'o'} = w]\!] \\
n_{w,e}^{\backslash d,o} &= \sum_{d'} \sum_{o'} [\![e_{d'o'} = e]\!][\![w_{d'o'} = w]\!][\![d \neq d' \vee o \neq o']\!] \\
n_{d,e} &= \sum_{o'} [\![e_{d,o'} = e]\!] \\
n_{d,e}^{\backslash d,o} &= \sum_{o' \neq o} [\![e_{d,o'} = e]\!]
\end{aligned}$$

Complete (and/or correct) the following update equations for Gibb's sampling:

$$\Pr(e|d, o, \ldots) \propto (\alpha + n_{d,e}^{\backslash d,o}) \frac{\beta + \underline{\qquad\qquad}}{|E_0|\underline{\quad\;} + n_e^{\backslash d,o}}$$

$$\phi_e(w) \propto \beta + \underline{\qquad\qquad}$$

(You may or may not choose to use the counts defined above.)

		4

The "$\ldots$" in $\Pr(e|d,o,\ldots)$ is to denote that the state of all random variables other than E_{do} are pinned, using counts that exclude position d,o. All w_{do} are always observed. First consider estimating θ^d from offsets $o' \neq o$ in document d (because the entity labels there are known). If we ignored the Dirichlet prior, then the only information about θ would be limited to document d itself, and we would estimate

$$\Pr(E_{do}=e|\ldots) = \frac{n_{d,e}^{\setminus d,o}}{\sum_{e'} n_{d,e'}^{\setminus d,o}}$$

Because Dirichlet is a conjugate prior for Multinomial, we get

$$\Pr(E_{do}=e|\text{ent's \& words except at } d,o) = \frac{\alpha + n_{d,e}^{\setminus d,o}}{\sum_{e'} \alpha + n_{d,e'}^{\setminus d,o}}$$

Now we add the information from w_{do} using Bayes' rule:

$$\begin{aligned}&\Pr(E_{do}=e|w_{do}, \text{ent's \& words except at } d,o)\\ &= \frac{\Pr(E_{do}=e|\text{ent's \& words except at } d,o)\Pr(w_{do}|e)}{\sum_{e'}\Pr(E_{do}=e'|\text{ent's \& words except at } d,o)\Pr(w_{do}|e')}\end{aligned}$$

From the plate diagram, w_{do} is drawn from Multi(ϕ), where ϕ is drawn from Dir(β). If V is the corpus vocabulary, then we have

$$\Pr(W_{do}=w|e,\beta) = \frac{\beta + n_{w,e}^{\setminus d,o}}{\sum_{w'}(\beta + n_{w',e}^{\setminus d,o})} = \frac{\beta + n_{w,e}^{\setminus d,o}}{|V|\beta + \sum_{w'} n_{w',e}^{\setminus d,o}} = \frac{\beta + n_{w,e}^{\setminus d,o}}{|V|\beta + n_{e}^{\setminus d,o}},$$

again because Dirichlet is a conjugate prior to multinomial. Combining the parts above,

$$\begin{aligned}&\Pr(E_{do}=e|w_{do}, \text{ent's \& words except at } d,o)\\ &= \frac{\dfrac{\alpha + n_{d,e}^{\setminus d,o}}{\sum_{e'}\alpha + n_{d,e'}^{\setminus d,o}} \quad \dfrac{\beta + n_{w,e}^{\setminus d,o}}{|V|\beta + n_{e}^{\setminus d,o}}}{\sum_{\hat{e}} \dfrac{\alpha + n_{d,\hat{e}}^{\setminus d,o}}{\sum_{e'}\alpha + n_{d,e'}^{\setminus d,o}} \quad \dfrac{\beta + n_{w,\hat{e}}^{\setminus d,o}}{|V|\beta + n_{\hat{e}}^{\setminus d,o}}}\end{aligned}$$

$\sum_{e'}\ldots$ cancels, and we get the simpler form

$$\begin{aligned}&\Pr(E_{do}=e|w_{do}, \text{ent's \& words except at } d,o)\\ &\propto \left(\alpha + n_{d,e}^{\setminus d,o}\right)\frac{\beta + n_{w,e}^{\setminus d,o}}{|V|\beta + n_{e}^{\setminus d,o}}\end{aligned}$$

Now coming to the update of $\phi_e(w)$. This time the entity and word at all positions are observed, so the MLE update would be

$$\phi_e(w) = \frac{\text{number of positions with } e \text{ and } w}{\text{number of positions with } e}$$

Accounting for the Dir(β) prior, we get

$$\phi_e(w) = \frac{\beta + n_{w,e}}{|V|\beta + n_e} \quad \text{i.e.,} \quad \phi_e(w) \propto \beta + n_{w,e}.$$

Throughout, when writing Pr($\bullet|\ldots$), we may have omitted some variables in the rhs "…" on which $\bullet$ is conditioned, if clear from context.

3(b) The problem with LDA for entity annotation is that it models the document as a bag of words and not a sequence, thereby missing the signal that words that help to disambiguate a mention are often close to it. Now we propose the following modified process "CE" (contiguous entity) to estimate ϕ by taking Markovian influence between the latent entities of adjacent word offsets. Each document d first fixes E_d, the set of entities to be mentioned in it. In training data, it is observable. To sample the latent "topic" (entity) of offset o, toss a fair coin. In case of heads, sample an entity from E_d uniformly at random (this can be made more sophisticated). In case of tails, copy the entity from the previous offset $o-1$. Modify the update equations from above and write it out in full below.

		4

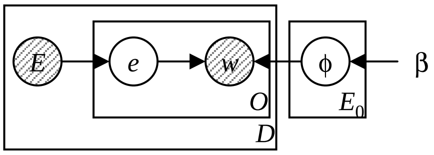
Pr($E_{do} = e|w_{do}$, ent's & words except at d, o) is modified to

$$\propto \begin{cases} \left(\frac{1}{2}[\![e = e_{d,o-1}]\!] + \frac{1}{2}\frac{1}{|E_d|}\right) \dfrac{\beta + n_{w,e}^{\setminus d,o}}{|V|\beta + n_e^{\setminus d,o}} & e \in E_d \\ 0 & e \notin E_d \end{cases}$$

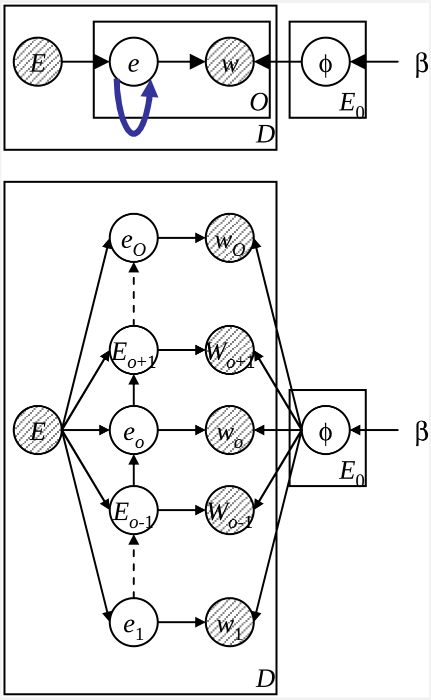
(Note that the 1/2 is not necessary if we also write "$\propto$".) Estimation of $\phi_e(w)$ remains the same as before.

3(c) There is one missing arrow in the modified plate diagram for CE below. Add it to the diagram. Also draw an offset-unrolled version of the diagram.

		2



There should be a self-loop at the e-node. The unrolled version is sketched below.



Total: 30