Natural Language Processing CSCI 4152/6509 — Lecture 10 Basic Probabilistic Models; P0 Topics Discussion

Instructors: Vlado Keselj
Time and date: 16:05 – 17:25, 9-Oct-2024
Location: Carleton Tupper Building Theatre C

Previous Lectures

- P0 Topics Discussion: P-01, P-03, P-04, P-05, P-06
- **Probabilistic approach to NLP**
- Logical vs. plausible reasoning
- Probability theory review
- Bayesian inference: generative models
- Probabilistic modeling:
    - random variables, random models
    - full and partial model configurations
    - computational tasks in probabilistic modeling

Joint Distribution Model

- Probability of each complete configuration is specified; i.e., the **joint probability distribution**:

$$\mathrm{P}(V_1{=}x_1, ..., V_n{=}x_n)$$

- If each variable can have m possible values, the model has m^n parameters
- The model is a large lookup table: For each full configuration $\mathbf{x} = (V_1{=}x_1, ..., V_n{=}x_n)$, a parameter $p_\mathbf{x}$ is specified such that

$$0 \le p_\mathbf{x} \le 1 \text{ and } \sum_\mathbf{x} p_\mathbf{x} = 1$$

Example: Spam Detection (Joint Distribution Model)

MLE — *Maximum Likelihood Estimation* of probabilities:

Free	*Caps*	*Spam*	Number of messages	p
Y	Y	Y	20	0.20
Y	Y	N	1	0.01
Y	N	Y	5	0.05
Y	N	N	0	0.00
N	Y	Y	20	0.20
N	Y	N	3	0.03
N	N	Y	2	0.02
N	N	N	49	0.49
		Total:	100	1.00

Computational Tasks in Joint Distribution Model:

1. Evaluation

- Evaluate the probability of a complete configuration $\mathbf{x} = (x_1, ..., x_n)$.
- Use a table lookup:

$$\mathrm{P}(V_1 = x_1, ..., V_n = x_n) = p_{(x_1, x_2, ..., x_n)}$$

- For example:

$$\mathrm{P}(\textit{Free} = Y, \textit{Caps} = N, \textit{Spam} = N) = 0.00$$

- This example illustrates the **sparse data problem**
- Inferred that the probability is zero since the configuration was not seen before.

2. Simulation (Joint Distribution Model)

2. Simulation (Joint Distribution Model)

- Simulation is performed by randomly selecting a configuration according to the probability distribution in the table
- Known as the "roulette wheel" method

1. Divide the interval $[0, 1]$ into subintervals of the lengths: p_1, p_2, $\ldots$, p_{m^n}: $I_1 = [0, p_1)$, $I_2 = [p_1, p_1 + p_2)$, $I_3 = [p_1 + p_2, p_1 + p_2 + p_3)$, $\ldots$ $I_{m^n} = [p_1 + p_2 + \ldots + p_{m^n-1}, 1)$
2. Generate a random number r from the interval $[0, 1)$
3. r will fall exactly into one of the above intervals, e.g.: $I_i = [p_1 + \ldots + p_{i-1}, p_1 + \ldots + p_{i-1} + p_i)$
4. Generate the configuration number i from the table
5. Repeat steps 2–4 for as many times as the number of configurations we need to generate

Joint Distribution Model: 3. Inference

3.a Marginalization

- Compute the probability of an *incomplete* configuration $\mathrm{P}(V_1 = x_1, ..., V_k = x_k)$, where $k < n$:

$$\mathrm{P}(V_1 = x_1, \ldots, V_k = x_k)$$
$$= \sum_{y_{k+1}} \cdots \sum_{y_n} \mathrm{P}(V_1 = x_1, \ldots, V_k = x_k, V_{k+1} = y_{k+1}, \ldots, V_n = y_n)$$
$$= \sum_{y_{k+1}} \cdots \sum_{y_n} p_{(x_1, \ldots, x_k, y_{k+1}, \ldots, y_n)}$$

- Implementation: iterate through the lookup table and accumulate probabilities for matching configurations

Joint Distribution Model: 3.b Conditioning

- Compute a conditional probability of assignments of some variables given the assignments of other variables; for example,

$$\mathrm{P}(V_1 = x_1, \ldots, V_k = x_k | V_{k+1} = y_1, ..., V_{k+l} = y_l)$$
$$= \frac{\mathrm{P}(V_1 = x_1, \ldots, V_k = x_k, V_{k+1} = y_1, ..., V_{k+l} = y_l)}{\mathrm{P}(V_{k+1} = y_1, ..., V_{k+l} = y_l)}$$

- This task can be reduced to two marginalization tasks
- If the configuration in the numerator happens to be a full configuration, that the task is even easier and reduces to one evaluation and one marginalization.

Joint Distribution Model: 3.c Completion

- Find the most probable completion $(y^*_{k+1}, ..., y^*_n)$ given a partial configuration $(x_1, ..., x_k)$.

$$\begin{aligned}
y^*_{k+1}, ..., y^*_n &= \arg\max_{y_{k+1},\ldots,y_n} \mathrm{P}(V_{k+1}=y_{k+1}, ..., V_n=y_n | V_1=x_1, ..., V_k=x_k) \\
&= \arg\max_{y_{k+1},\ldots,y_n} \frac{\mathrm{P}(V_1=x_1, ..., V_k=x_k, V_{k+1}=y_{k+1}, ..., V_n=y_n)}{\mathrm{P}(V_1=x_1, ..., V_k=x_k)} \\
&= \arg\max_{y_{k+1},\ldots,y_n} \mathrm{P}(V_1=x_1, \ldots, V_k=x_k, V_{k+1}=y_{k+1}, ..., V_n=y_n) \\
&= \arg\max_{y_{k+1},\ldots,y_n} p_{(x_1,\ldots,x_k,y_{k+1},\ldots,y_n)}
\end{aligned}$$

- Implementation: search through the model table, and from all configurations that satisfy assignments in the partial configuration, chose the one with maximal probability.

Joint Distribution Model: 4. Learning

- Estimate the parameters in the model based on given data
- Use **Maximum Likelihood Estimation** (MLE)
- Count all full configurations, divide the count by the total number of configurations, and fill the table:

$$p_{(x_1,\ldots,x_n)} = \frac{\#(V_1\!=\!x_1,\ldots,V_n\!=\!x_n)}{\#(*,\ldots,*)}$$

- With a large number of variables the data size easily becomes insufficient and we get many zero probabilities — **sparse data problem**

Drawbacks of Joint Distribution Model

- memory cost to store table,
- running-time cost to do summations, and
- the sparse data problem in learning (i.e., training).

Other probability models are found by specifying specialized joint distributions, which satisfy certain independence assumptions.
The goal is to impose structure on joint distribution $\mathrm{P}(V_1 = x_1, ..., V_n = x_n)$. One key tool for imposing structure is variable independence.

Fully Independent Model

- Assumption: all variables are independent

$$\mathrm{P}(V_1\!=\!x_1, ..., V_n\!=\!x_n) = \mathrm{P}(V_1\!=\!x_1)\cdots\mathrm{P}(V_n\!=\!x_n).$$

- Efficient model with a small number of parameters: $O(nm)$
- Drawback: usually a too strong assumption
- Fully independent model for the Spam example:

$$\mathrm{P}(\textit{Free}, \textit{Caps}, \textit{Spam}) = \mathrm{P}(\textit{Free}) \cdot \mathrm{P}(\textit{Caps}) \cdot \mathrm{P}(\textit{Spam})$$

Fully Independent Model: [4.] Learning

Spam example:

Free	$\mathrm{P}(\textit{Free})$
Y	$\frac{20+1+5+0}{100} = 0.26$
N	$\frac{20+3+2+49}{100} = 0.74$

and similarly,

Caps	$\mathrm{P}(\textit{Caps})$
Y	$\frac{20+1+20+3}{100} = 0.44$
N	$\frac{5+0+2+49}{100} = 0.56$

and

Spam	$\mathrm{P}(\textit{Spam})$
Y	$\frac{20+5+20+2}{100} = 0.47$
N	$\frac{1+0+3+49}{100} = 0.53$

Hence, in this model any message is a spam with probability 0.47, no matter what the values of *Caps* and *Free* are.

Evaluation Example

As an example of evaluation, the probability of configuration $(\mathit{Caps} = Y, \mathit{Free} = N, \mathit{Spam} = N)$ in the fully independent model is:

$$
\begin{aligned}
&\mathrm{P}(\mathit{Free} = Y, \mathit{Caps} = N, \mathit{Spam} = N) = \\
&\quad = \mathrm{P}(\mathit{Free} = Y) \cdot \mathrm{P}(\mathit{Caps} = N) \cdot \mathrm{P}(\mathit{Spam} = N) = \\
&\quad = 0.26 \cdot 0.56 \cdot 0.53 \\
&\quad = 0.077168 \approx 0.08
\end{aligned}
$$

Fully Independent Model: 2. Simulation

- For $j = 1, ..., n$, independently draw x_j according to $\mathrm{P}(V_j = x_j)$ using "roulette wheel" for one variable
- Conjoin $(x_1, ..., x_n)$ to form a complete configuration.

3. Inference in Fully Independent Model

3.a Marginalization in Fully Independent Model
The probability of a partial configuration
$(V_1 = x_1, \ldots, V_k = x_k)$ is

$$P(V_1 = x_1, \ldots, V_k = x_k) = P(V_1 = x_1) \cdot \ldots \cdot P(V_k = x_k)$$

This formula can be obvious, but it can also be derived.

Derivation of Marginalization Formula

$$\mathrm{P}(V_1 = x_1, \ldots, V_k = x_k) = \sum_{y_{k+1}} \cdots \sum_{y_n} \mathrm{P}(V_1 = x_1, \ldots, V_k = x_k, V_{k+1} = y_{k+1}, \ldots, V_n = y_n)$$

$$= \sum_{y_{k+1}} \cdots \sum_{y_n} \mathrm{P}(V_1 = x_1) \cdots \mathrm{P}(V_k = x_k) \mathrm{P}(V_{k+1} = y_{k+1}) \cdots \mathrm{P}(V_n = y_n)$$

$$= \mathrm{P}(V_1 = x_1) \cdots \mathrm{P}(V_k = x_k) \left[\sum_{y_{k+1}} \mathrm{P}(V_{k+1} = y_{k+1}) \left[\sum_{y_{k+2}} \cdots \left[\sum_{y_n} \mathrm{P}(V_n = y_n) \right] \right] \right]$$

$$= \mathrm{P}(V_1 = x_1) \cdots \mathrm{P}(V_k = x_k) \left[\sum_{y_{k+1}} \mathrm{P}(V_{k+1} = y_{k+1}) \right] \cdots \left[\sum_{y_n} \mathrm{P}(V_n = y_n) \right]$$

$$= \mathrm{P}(V_1 = x_1) \cdots \mathrm{P}(V_k = x_k)$$

A Note on Sum-Product Computation

$$\begin{aligned}
\sum_a \sum_b f(a)g(b) &= \sum_a f(a) \left(\sum_b g(b) \right) \\
&\quad \text{(because } f(a) \text{ is a constant for summation over } b\text{)} \\
&= \left(\sum_b g(b) \right) \cdot \left(\sum_a f(a) \right) \\
&\quad \text{(because } \sum_b g(b) \text{ is a constant for sumation over } a\text{)} \\
&= \left(\sum_a f(a) \right) \cdot \left(\sum_b g(b) \right)
\end{aligned}$$

Similar Note for Max-Product Computation

If we assume that $f(a) \geq 0$ and $g(b) \geq 0$, the same rule applies for $\max_a$ and $\max_b$:

$$\begin{aligned}
& \max_a \max_b f(a)g(b) = \\
= \quad & \max_a f(a) \left(\max_b g(b) \right) \\
& \text{(because } f(a) \text{ is a constant for maximization over } b\text{)} \\
= \quad & \left(\max_b g(b) \right) \cdot \left(\max_a f(a) \right) \\
& \text{(because } \max_b g(b) \text{ is a constant for maximization over } a\text{)} \\
= \quad & \left(\max_a f(a) \right) \cdot \left(\max_b g(b) \right)
\end{aligned}$$

P0 Topics Discussion (2)

- Discussion of individual projects as proposed in P0 submissions
- Projects discussed: P-02