# CSCI 4152/6509 — Natural Language Processing *14-Oct-2009*

## Lecture 14: Fully Independent Model

Room: FASS 2176
Time: 11:35 – 12:25

**Previous Lecture**

- Probabilistic modeling:
  - random variables,
  - random configurations,
  - computational tasks in probabilistic modeling,
- Spam detection example,
- Joint distribution model,
- Drawbacks of joint distribution model

---

## 8.5 Fully Independent Model

In a fully independent model we assume that all variables are independent, i.e.,

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

which is the evaluation formula, (**1. Evaluation**)
It is an efficient model with a small number of parameters: $O(nm)$
Drawback: usually a too strong assumption
Translated into the spam example:

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

This yields a very restricted form of joint distribution where we can represent each component distribution separately. For a random variable $V_j$, one can represent $\mathrm{P}(V_j = x)$ by a lookup table with $m$ parameters (minus one constraint). Let $p_{j,x}$ denote the probability $V_j$ takes on value $x$. That is, $p_{j,x} = \mathrm{P}(V_j = x)$. These numbers are positive and satisfy the constraint $\sum_{x=1}^{m} p_{j,x} = 1$ for each $j$. Thus, the joint distribution over $V_1, ..., V_n$ can be represented by $n \times m$ positive numbers minus $n$ constraints. The previous tasks (simulation, evaluation, and inference) now become almost trivial. Admittedly this is a silly model as far as real applications go, but it clearly demonstrates the benefits of structure (in its most extreme form).

### Example: Spam Detection (continued)

The fully independent model is almost useless in our spam detection example because it assumes that the three random variables: *Caps, Free,* and *Spam* are independent. In other words, its assumption is that knowing whether a message has a capitalized subject or contains the word ‘Free’ in the subject cannot help us in determining whether the message is spam or not, which is not in accordance with our earlier assumption.
Anyway, let us see what happens when we apply the fully independent model to our example. From the training data:

| *Free* | *Caps* | *Spam* | Number of messages |
|---|---|---|---|
| Y | Y | Y | 20 |
| Y | Y | N | 1 |
| Y | N | Y | 5 |
| Y | N | N | 0 |
| N | Y | Y | 20 |
| N | Y | N | 3 |
| N | N | Y | 2 |
| N | N | N | 49 |
| | | Total: | 100 |

we generate the following probability tables of independent variables:

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

and similarly,

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

and

| *Spam* | P(*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. This is example of MLE **Learning** (computational task 4.).

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

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

### 2. Simulation

For $j = 1, ..., n$, independently draw $x_j$ according to $\mathrm{P}(V_j = x_j)$ (using the lookup table representation). Conjoin $(x_1, ..., x_n)$ to form a complete configuration.

### 3. Inference

#### 3.a Marginalization

The probability of a partial configuration $(x_1, \ldots, x_k)$ is

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

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

**Derivation of Marginalization Formula**

$$
\begin{aligned}
\mathrm{P}(V_1\!=\!x_1,...,V_k\!=\!x_k) &= \sum_{y_{k+1}}\cdots\sum_{y_n}\mathrm{P}(V_1\!=\!x_1,...,V_k\!=\!x_k,V_{k+1}\!=\!y_{k+1},...,V_n\!=\!y_n)\\
&= \sum_{y_{k+1}}\cdots\sum_{y_n}\mathrm{P}(V_1\!=\!x_1)\cdots\mathrm{P}(V_k\!=\!x_k)\mathrm{P}(V_{k+1}\!=\!y_{k+1})\cdots\mathrm{P}(V_n\!=\!y_n)\\
&= \mathrm{P}(V_1\!=\!x_1)\cdots\mathrm{P}(V_k\!=\!x_k)\left[\sum_{y_{k+1}}\mathrm{P}(V_{k+1}\!=\!y_{k+1})\left[\sum_{y_{k+2}}\cdots\left[\sum_{y_n}\mathrm{P}(V_n\!=\!y_n)\right]\right]\right]\\
&= \mathrm{P}(V_1\!=\!x_1)\cdots\mathrm{P}(V_k\!=\!x_k)\left[\sum_{y_{k+1}}\mathrm{P}(V_{k+1}\!=\!y_{k+1})\right]\cdots\left[\sum_{y_n}\mathrm{P}(V_n\!=\!y_n)\right]\\
&= \mathrm{P}(V_1\!=\!x_1)\cdots\mathrm{P}(V_k\!=\!x_k)
\end{aligned}
$$

Only have to lookup and multiply $k$ numbers.

**Note**

It is important to note a general rule which we used to separate summations in the above tasks of Marginalization and Completion: If $a$ and $b$ are two variables, and $f(a)$ and $g(b)$ are two functions, such that $f(a)$ does not depend on $b$ and $g(b)$ does not depend on $a$, then:

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

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

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

### 3.b Conditioning

$$\begin{aligned}
&\mathrm{P}(V_{k+1}{=}y_{k+1},...,V_n{=}y_n|V_1{=}x_1,...,V_k{=}x_k)\\
&= \frac{\mathrm{P}(V_1{=}x_1,...,V_k{=}x_k,V_{k+1}{=}y_{k+1},...,V_n{=}y_n)}{\mathrm{P}(V_1{=}x_1,\ldots,V_k{=}x_k)}\\
&= \frac{\mathrm{P}(V_1{=}x_1)\cdots\mathrm{P}(V_k{=}x_k)\mathrm{P}(V_{k+1}{=}y_{k+1})\cdots\mathrm{P}(V_n{=}y_n)}{\mathrm{P}(V_1{=}x_1)\cdots\mathrm{P}(V_k{=}x_k)}\\
&= \mathrm{P}(V_{k+1}{=}y_{k+1})\cdots\mathrm{P}(V_n{=}y_n)
\end{aligned}$$

Only have to lookup and multiply $n-k$ numbers.

### 3.c Completion

$$\begin{aligned}
y^*_{k+1},...,y^*_n &= \operatorname*{arg\,max}_{y_{k+1},\ldots,y_n} \mathrm{P}(V_{k+1}{=}y_{k+1},...,V_n{=}y_n|V_1{=}x_1,...,V_k{=}x_k)\\
&= \operatorname*{arg\,max}_{y_{k+1},\ldots,y_n} \mathrm{P}(V_{k+1}{=}y_{k+1})\cdots\mathrm{P}(V_n{=}y_n)\\
&= \operatorname*{arg\,max}_{y_{k+1}} \mathrm{P}(V_{k+1}{=}y_{k+1})\left[\operatorname*{arg\,max}_{y_{k+2}}\cdots\left[\operatorname*{arg\,max}_{y_n}\mathrm{P}(V_n{=}y_n)\right]\right]
\end{aligned}$$

(Since $\max$ and arg max distributes over product just like sum.
That is, $\max_i ax_i = a\max_i x_i$ (for $a, x_i \geq 0$)
just like $\sum_i ax_i = a\sum_i x_i$.)

$$\begin{aligned}
&= \left[\operatorname*{arg\,max}_{y_{k+1}}\mathrm{P}(V_{k+1}{=}y_{k+1})\right]\cdots\left[\operatorname*{arg\,max}_{y_n}\mathrm{P}(V_n{=}y_n)\right]\\
&= \left[\operatorname*{arg\,max}_{y_{k+1}} p_{k+1,y_{k+1}}\right]\cdots\left[\operatorname*{arg\,max}_{y_n} p_{n,y_n}\right]
\end{aligned}$$

Only have to search through $m$ possible completions for each of the $n-k$ variables separately.

### Joint Distribution Model vs. Fully Independent Model

The Fully Independent Model addresses the previous issues with the joint distribution model, but it suffers from a too strong assumption and too little structure, so it usually does not model accurately the real relationships among variables.

**Structured probability models** are a compromise solution between previous two models. Structured probability models are more efficient than the joint distribution model and they address the issue of the sparse training data, and in the same time they model important dependencies among random variables.

One of the simplest models of this kind is the Naïve Bayes Model.