Natural Language Processing
CSCI 4152/6509 — Lecture 9
Introduction to Probabilistic Modeling

Instructors: Vlado Keselj
Time and date: 14:35 – 15:55, 23-Oct-2025
Location: Studley LSC-Psychology P5260

Previous Lecture

- Evaluation methods for Text Classification:
  - underfitting and overfitting
  - training error, train and test, n-fold cross-validation
- Text Clustering
- Discussion about evaluation methods for classifiers
- Similarity-based Text Classification
- CNG classification method
- Edit distance:
  - introduction, properties, dynamic programming approach, example, algorithm

Part III: Probabilistic Approach to NLP

Logical versus Plausible Reasoning

- As a part of AI (Artificial Intelligence), NLP follows two main approaches to *computer reasoning*, or *computer inference:*

1. logical reasoning
   - known also as classical, symbolic, knowledge-based AI
   - *monotonic:* once conclusion drawn, never retracted
   - *certain:* conclusions certain, given assumptions
2. plausible reasoning
   - examples: probabilistic, fuzzy logic, neural networks
   - *non-monotonic*
   - *uncertain*

Plausible Reasoning

- How to combine ambiguous, incomplete, and contradicting evidence to draw reasonable conclusions?
- Typical approach: make plausible inference of some hidden structure from observations
- Examples:

Observations (input)		Hidden Structure (output)
symptoms	$\rightarrow$	illness
pixel matrix	$\rightarrow$	object, relations
speech signal	$\rightarrow$	phonemes, words
word sequence	$\rightarrow$	meaning
sentence	$\rightarrow$	parse tree
word sequence	$\rightarrow$	POS tags, names, entities
words in e-mail Subject:	$\rightarrow$	Is message spam? Yes/No
text	$\rightarrow$	text category (class)

Probabilistic NLP as a Plausible Reasoning Approach

- Regular expressions and finite automata are example of logical or knowledge-based approach to NLP
- Plausible approaches to NLP:
  1. Probabilistic: use of Theory of Probability, also known as stochastic or statistical NLP
  - Alternative plausible approaches, examples:
  2. neural networks,
  3. kernel methods,
  4. fuzzy logic, fuzzy sets,
  5. Dempster-Shafer theory
  6. rough sets,
  7. default logic, ...

Review of Basics of Probability Theory

- You should have this background from previous courses; this is just a review,
  - discussed a bit in the textbook: [JM] 5.5, and [MS] 2.1
- Simple event or basic outcome
  - e.g., rolling a die, choosing a letter
- *Event space:* the set of all outcomes, usually denoted Ω
- *Event or outcome* is a set of simple events or basic outcomes
- In other words event is any subset of Ω; i.e., $A \subseteq \Omega$
- Each event is associated with a probability, which is a number between 0 and 1, inclusive: $0 \leq \mathrm{P}(A) \leq 1$

Probability Examples

- $\mathrm{P}(\text{“rolling a 6 with a die”}) = 1/6$
- Choosing a letter of English alphabet:
  - If we choose uniformly: $\mathrm{P}(\text{‘a’}) = 1/26 \approx 0.04$
  - Choosing from a text: $\mathrm{P}(\text{‘a’}) \approx 0.08$
  - Remember our output from “Tom Sawyer”:

```
35697 0.1204 e
28897 0.0974 t
23528 0.0793 a
23264 0.0784 o
20200 0.0681 n
...
```

Probability Axioms

- **(Nonnegativity)** $\mathrm{P}(A) \geq 0$, for any event A
- **(Additivity)** for disjoint events A and B, i.e., if $A, B \subset \Omega$ and $A \cap B = \emptyset$, then
  $\mathrm{P}(A \cup B) = \mathrm{P}(A) + \mathrm{P}(B)$
  or, more generally,
  $\mathrm{P}(A_1 \cup A_2 \cup \ldots) = \mathrm{P}(A_1) + \mathrm{P}(A_2) + \ldots$
- **(Normalization)** $\mathrm{P}(\Omega) = 1$, where Ω is the entire sample space.
- Some consequences of the above axioms are:
  $\mathrm{P}(\emptyset) = 0$ and $\mathrm{P}(\Omega - A) = 1 - \mathrm{P}(A)$

Independent and Dependent Events

- Independent events A and B (definition):
  $\mathrm{P}(A, B) = \mathrm{P}(A) \cdot \mathrm{P}(B)$
- Use of comma in: $\mathrm{P}(A, B) = \mathrm{P}(A \cap B)$
- Example: choosing two letters in text
  1. Choosing independently:
     $\mathrm{P}(\text{'t'}) = 0.1, \mathrm{P}(\text{'h'}) = 0.07, \mathrm{P}(\text{'t'}, \text{'h'}) = 0.007$
  2. Choosing two consecutive letters (dependent events):
     $\mathrm{P}(\text{'t'}, \text{'h'}) = 0.04$

Conditional Probability

- Conditional probability

$$\mathrm{P}(A|B) = \frac{\mathrm{P}(A,B)}{\mathrm{P}(B)}$$

- Expressing independency using conditional probability

  Two events A are B are independent if and only if:

$$\mathrm{P}(A|B) = \mathrm{P}(A)$$

  This is an alternative definition of independent events.

Annotation with More Events

- There is a bit of flexibility in using notation; e.g.,
- $\mathrm{P}(A, B, C) = \mathrm{P}(A \cap B \cap C)$
- $\mathrm{P}(A|B, C) = \mathrm{P}(A|B \cap C)$
- $\mathrm{P}(A, B, C|D, E, F) = \mathrm{P}(A \cap B \cap C|D \cap E \cap F)$
- and so on.
- Three independent events: $\mathrm{P}(A, B, C) = \mathrm{P}(A)\mathrm{P}(B)\mathrm{P}(C)$
- Conditionally independent events

$$\mathrm{P}(A, B|C) = \mathrm{P}(A|C) \cdot \mathrm{P}(B|C)$$

Bayes' Theorem

- Bayes' theorem (one form):

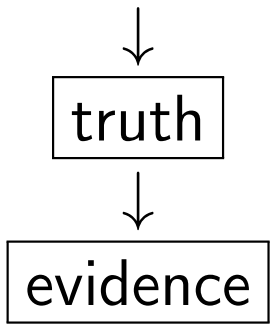
$$\mathrm{P}(A|B) = \frac{\mathrm{P}(B|A) \cdot \mathrm{P}(A)}{\mathrm{P}(B)}$$

- The second form is based on breaking the set B into disjoint sets $B = A_1 \cup A_2 \cup \ldots$:

$$\mathrm{P}(A_i|B) = \frac{\mathrm{P}(B|A_i) \cdot \mathrm{P}(A_i)}{\mathrm{P}(B)} = \frac{\mathrm{P}(B|A_i) \cdot \mathrm{P}(A_i)}{\sum_i \mathrm{P}(B|A_i)\mathrm{P}(A_i)}$$

Bayesian Inference and Generative Models

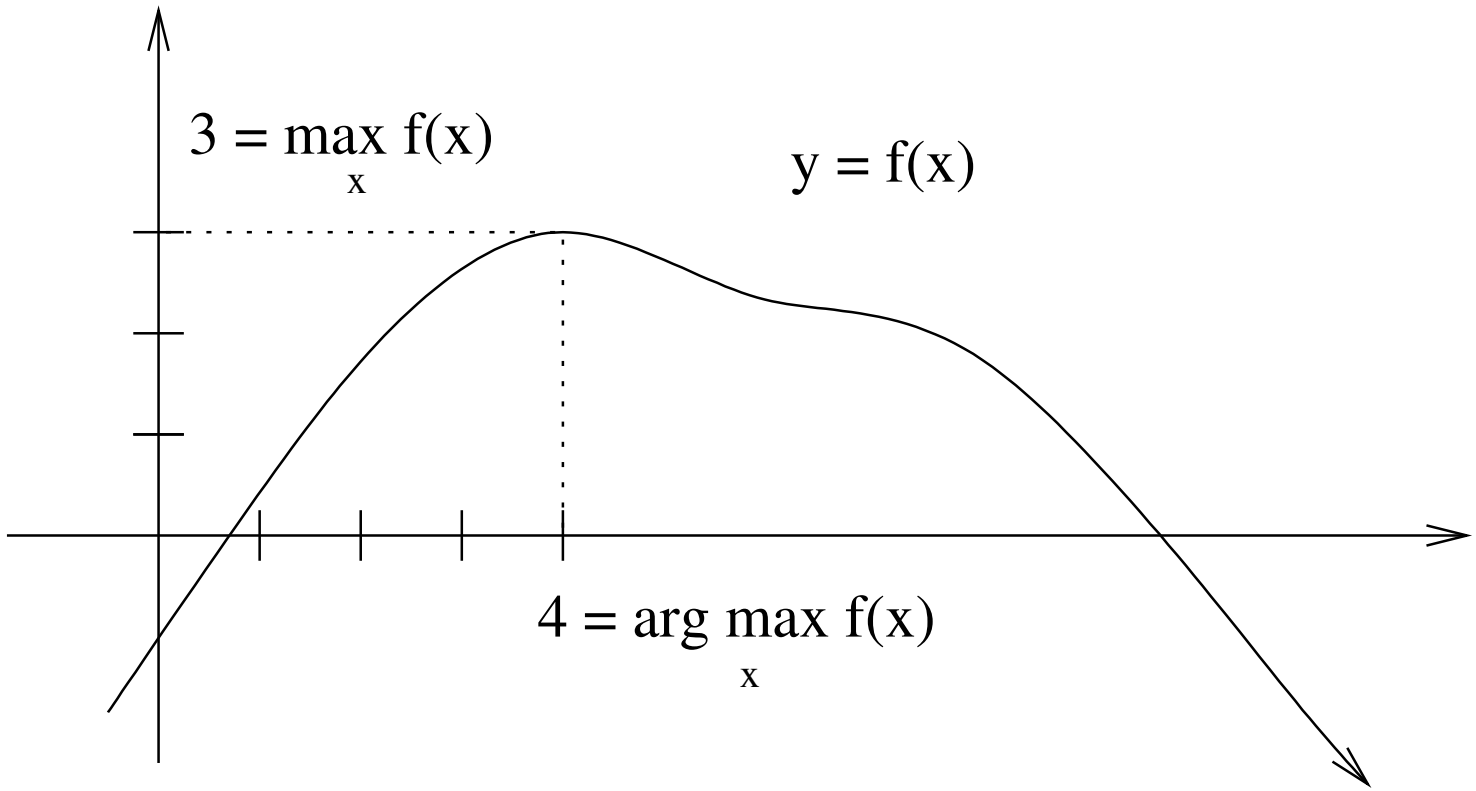
- We will use Bayesian Inference on Generative Models
- Generative Models, also known as Forward Generative Models
- One way of representing knowledge with a probabilistic model



world
↓
sensor
↓
observed measurement

Notation Remark: max and argmax

- $\max$ is the maximum value of a function
- $\arg\max$ is an argument value for which function achieves the maximum



Bayesian Inference: Using Bayes' Theorem

- Bayesian inference is a principle of combining evidence

$$
\begin{aligned}
\text{conclusion} &= \underset{\text{possible truth}}{\arg\max}\ P(\text{possible truth}|\text{evidence}) \\
&= \underset{\text{possible truth}}{\arg\max}\ \frac{P(\text{evidence}|\text{possible truth})P(\text{possible truth})}{P(\text{evidence})} \\
&= \underset{\text{possible truth}}{\arg\max}\ P(\text{evidence}|\text{possible truth})P(\text{possible truth})
\end{aligned}
$$

- application to speech recognition: acoustic model and language model

Bayesian Inference: Speech Recognition Example

- evidence $\rightarrow$ sound
- possible truth $\rightarrow$ utterance (words spoken)
- our best guess about utterance $\rightarrow$ utterance*

$$
\begin{aligned}
\text{utterance}^* &= \underset{\text{all utterances}}{\arg\max}\ P(\text{utterance}|\text{sound}) \\
&= \underset{\text{all utterances}}{\arg\max}\ \frac{P(\text{sound}|\text{utterance})P(\text{utterance})}{P(\text{sound})} \\
&= \underset{\text{utterance}}{\arg\max}\ P(\text{sound}|\text{utterance})P(\text{utterance})
\end{aligned}
$$

Probabilistic Modeling

- How do we create and use a probabilistic model?
- Model elements:
  - Random variables
  - Model configuration (Random configuration)
  - Variable dependencies
  - Model parameters
- Computational tasks

Random Variables

- Random variable V, defining an event as $V = x$ for some value x from a domain of values D; i.e., $x \in D$
- $V = x$ is usually not a **basic** event due to having more variables
- An event with two random variables: $V_1 = x_1, V_2 = x_2$
- Multiple random variables: $\mathbf{V} = (V_1, V_2, ..., V_n)$

Model Configuration (Random Configuration)

- **Full Configuration:** If a model has n random variables, then a Full Model Configuration is an assignment of all the variables:

$$V_1 = x_1, V_2 = x_2, \ldots, V_n = x_n$$

- **Partial configuration:** only some variables are assigned, e.g.:

$$V_1 = x_1, V_2 = x_2, \ldots, V_k = x_k \quad (k < n)$$

Probabilistic Modeling in NLP

Probabilistic Modeling in NLP is a general framework for modeling NLP problems using random variables, random configurations, and an effective ways to reason about probabilities of these configurations.

Variable Independence and Dependence

- Random variables V_1 and V_2 are *independent* if $\mathrm{P}(V_1 = x_1, V_2 = x_2) = \mathrm{P}(V_1 = x_1)\mathrm{P}(V_2 = x_2)$ for all x_1, x_2
- or expressed in a different way: $\mathrm{P}(V_1 = x_1 | V_2 = x_2) = \mathrm{P}(V_1 = x_1)$ for all x_1, x_2, x_3.
- Random variables V_1 and V_2 are *conditionally independent given* V_3 if, for all x_1, x_2, x_3: $\mathrm{P}(V_1 = x_1, V_2 = x_2 | V_3 = x_3) =$ $\mathrm{P}(V_1 = x_1 | V_3 = x_3)\mathrm{P}(V_2 = x_2 | V_3 = x_3)$
- or $\mathrm{P}(V_1 = x_1 | V_2 = x_2, V_3 = x_3) = \mathrm{P}(V_1 = x_1 | V_3 = x_3)$

Computational Tasks in Probabilistic Modeling

1. Evaluation: compute probability of a complete configuration
2. Simulation: generate random configurations
3. Inference: has the following sub-tasks:
    - 3.a Marginalization: computing probability of a partial configuration,
    - 3.b Conditioning: computing conditional probability of a completion given an observation,
    - 3.c Completion: finding the most probable completion, given an observation
4. Learning: learning parameters of a model from data.

Illustrative Example: Spam Detection

- the problem of spam detection
- a probabilistic model for spam detection; random variables:

$$
\begin{aligned}
\mathit{Caps} &= \text{'Y' if the message subject line does not contain lowercase letter, 'N' otherwise,} \\
\mathit{Free} &= \text{'Y' if the word 'free' appears in the message subject line (letter case is ignored), 'N' otherwise, and} \\
\mathit{Spam} &= \text{'Y' if the message is spam, and 'N' otherwise.}
\end{aligned}
$$

- one random configuration represents one e-mail message

Random Sample

- Data based on sample of 100 email messages

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

What are examples of computational tasks in this example?

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

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

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:

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

- 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 $k < n$:

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

- 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,

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

- 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 $(x_1, ..., x_k)$.

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

- 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:

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

Fully Independent Model: [4.] Learning

Spam example:

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.

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:

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

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

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

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

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

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

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

A Note on Sum-Product Computation

$$\sum_a \sum_b f(a)g(b) = \sum_a f(a) \left(\sum_b g(b) \right)$$

(because $f(a)$ is a constant for summation over b)

$$= \left(\sum_b g(b) \right) \cdot \left(\sum_a f(a) \right)$$

(because $\sum_b g(b)$ is a constant for sumation over a)

$$= \left(\sum_a f(a) \right) \cdot \left(\sum_b g(b) \right)$$

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 $\max_b$:

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

3.c Completion in Fully Independent Model

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

Joint Distribution Model vs. Fully Independent Model

- Fully Independent Model addresses some issues of the Joint Distribution Model
- Efficient and small number of parameters
- However: too strong assumption, no structure
- Too trivial to be usable
- Better method: Structured probability models
  - compromise between no dependence and too much dependence