CSCI 4152/6509 — Natural Language Processing

9-Oct-2009

Lecture 13: Joint Distribution Model

Room: FASS 2176
Time: 11:35 – 12:25

Previous Lecture

- NLP Anthology `http://aclweb.org/anthology-new/`,
- CNG classification method,
- Elements of probability theory,
- Generative models,
- Bayesian inference

8.2 Probabilistic Model

Random variables

We assume that we have a set of n random variables that capture an outcome in our model:

$$\mathbf{V} = (V_1, V_2, ..., V_n)$$

They may be observable variables, or hidden variables. Each variable can be assigned a value from a finite set of different values. We denote these values as $\{x_1, x_2, \ldots, x_m\}$.

Random configuration

A tuple of values, i.e., a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, where each value is assigned to a variable: $V_1 = x_1$, $V_2 = x_2, \ldots$ is called a **random configuration.**

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

In modelling our problem we assume that a sequence of configurations $\mathbf{x}^{(1)}, ..., \mathbf{x}^{(t)}$ is drawn from some random source:

$$\begin{array}{rcl} \mathbf{x}^{(1)} & = & (x_{11}, \quad x_{12}, \quad \ldots \quad x_{1n}) \\ \mathbf{x}^{(2)} & = & (x_{21}, \quad x_{22}, \quad \ldots \quad x_{2n}) \\ & \vdots & \\ \mathbf{x}^{(t)} & = & (x_{t1}, \quad x_{t2}, \quad \ldots \quad x_{tn}) \end{array}$$

Again, we assume a fixed number n of components in each configuration, and assume values x_{ij} are from a finite set $\{x_1, x_2, \ldots, x_m\}$.

Probabilistic modelling in NLP can be described as a general framework for modelling NLP problems using random variables, random configurations, and finding effective ways of reasoning about probabilities of these configurations.

8.3 Computational Tasks in Probabilistic Modelling

1. Evaluation: compute probability of a complete configuration

2. Simulation: generate random configurations

Simulation is also referred to as *generation,* or *sampling.*

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.

Let us use an example to illustrate this:

Example: Spam Detection
The problem of spam detection in e-mail is the problem of automatically detecting whether an arbitrary e-mail message is spam or not. In a toy model, we assume that we can detect whether a message is spam or not relying only on the fact whether the 'Subject:' header of the message is capitalized (i.e., completely written in uppercase letters) and whether the 'Subject:' header contains the word 'free' (uppercase or lowercase). For example, "NEW MORTGAGE RATE" is likely the subject of a spam message, as well as "Money for Free," "FREE lunch," etc.
Hence, our model is based on the following three random variables and each of them gets one of two values Y (for Yes) or N (for No):

Caps	=	'Y' if the message subject line does not contain lowercase letter, 'N' otherwise,
Free	=	'Y' if the word 'free' appears in the message subject line (letter case is ignored), 'N' otherwise, and
Spam	=	'Y' if the message is spam, and 'N' otherwise.

In order to learn what happens in real-world, we open our mailbox, which serves as our random source, randomly select 100 messages and count how many times each configuration appears.
We might obtain the following table:

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

Let us consider our first, straightforward model, called **Joint Distribution Model.**

8.4 Joint Distribution Model

In the Joint Distribution Model, we specify the complete **joint probability distribution,** i.e., the probability of each complete configuration $\mathbf{x} = (x_1, ..., x_n)$:

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

In general, we need m^n parameters (minus one constraint) to specify an arbitrary joint distribution on n random variables with m values. One could represent this by a lookup table $p_{\mathbf{x}^{(1)}}, p_{\mathbf{x}^{(2)}}, \ldots, p_{\mathbf{x}^{(m^n)}}$, where $p_{\mathbf{x}^{(\ell)}}$ gives the probability that the random variables jointly take on configuration $\mathbf{x}^{(\ell)}$; that is, $p_{\mathbf{x}^{(\ell)}} = \mathrm{P}(\mathbf{V} = \mathbf{x}^{(\ell)})$. These numbers are positive and satisfy the constraint that $\sum_{\ell=1}^{m^n} p_{\mathbf{x}^{(\ell)}} = 1$.

Example: Spam Detection (continued)

To estimate the joint distribution in our spam detection example, we can simply divide the number of message for each configuration with the total number of messages:

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

Estimating probabilities in this way is known as *Maximum Likelihood Estimation* (MLE), since it can be shown that in this way the probability $\mathrm{P}(T|M)$, where T is our training data and M is the model, is maximized in terms of M.

8.4.1 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 a drawback of the full joint distribution model: the **sparse data problem.**
We concluded that the probability of this configuration is 0, i.e., that it is impossible, based on the fact that we have not seen it before. (!?)
However, not seeing a configuration does not necessarily mean that it cannot appear in the future.

8.4.2 Simulation

Draw a complete configuration $\mathbf{x}$ according to the joint distribution. Given the lookup table representation, one could just compute the cumulative value of the $p_{\mathbf{x}^{(\ell)}}$'s, draw a random number p between 0 and 1, and select the configuration $\mathbf{x}^{(\ell)}$ whose cumulative probability interval contains p.

8.4.3 Inference

3.a *Marginalization.*
Compute the probability of an *incomplete* configuration $\mathrm{P}(X_1 = x_1, ..., X_k = x_k)$, where $k < n$:

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

We need to be able to evaluate complete configurations and then sum over m^{n-k} possible completions, where m is the number of elements in the domain of $y_{k+1}, \ldots, y_n$.

3.b *Conditioning*.
Compute the conditional probability of a possible completion $(y_{k+1}, ..., y_n)$ given an incomplete configuration $(x_1, ..., x_k)$.

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

Need to evaluate a complete configuration and then divide by a marginal sum.

3.c *Completion*.
Find the most probable completion $(y^*_{k+1}, ..., y^*_n)$ given an incomplete configuration $(x_1, ..., x_k)$.

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

Have to search through all m^{n-k} possible completions and evaluate each complete configuration to find the maximum.

8.4.4 Learning

Estimate probabilities by counting, i.e., using **maximum likelihood estimation.**
In the spam example, we simply calculated the table by dividing counts with the total number of samples.

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.