11 Probability Modeling

Random configuration

Imagine a sequence of "configurations" $\mathbf{x}^{(1)}, ..., \mathbf{x}^{(t)}$ is drawn from some random source:

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

Here we assume a fixed number (n) of components in each configuration, and assume values x_{ij} are from a finite set; e.g., $x_{ij} \in \{1, 2, ..., m\}$.

Random variables

$$\mathbf{X} = X_1, X_2, ..., X_n$$

11.1 Computational tasks

Representation

Simulation Generate random configurations

Evaluation Compute probability of a complete configuration

Marginalization Compute probability of a partial configuration

Conditioning Compute conditional probability of completion given a partial observation

Completion Find most probable completion of partial observation

Learning Estimate parameters (will be covered later in the course)

11.2 Joint probability distribution

A joint probability distribution $\mathrm{P}(X_1 = x_1, ..., X_n = x_n)$ specifies the probability of each complete configuration $\mathbf{x} = (x_1, ..., x_n)$. In general it takes 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 $\theta_{\mathbf{x}^{(1)}}, \theta_{\mathbf{x}^{(2)}}, ..., \theta_{\mathbf{x}^{(V^n)}}$; where $\theta_{\mathbf{x}^{(\ell)}}$ gives the probability that the random variables jointly take on configuration $\mathbf{x}^{(\ell)}$. That is, $\theta_{\mathbf{x}^{(\ell)}} = \mathrm{P}(\mathbf{X} = \mathbf{x}^{(\ell)})$. These numbers are positive and satisfy the constraint that $\sum_{\ell=1}^{V^n} \theta_{\mathbf{x}^{(\ell)}} = 1$.

Probability models are ways of representing/specifying *specialized* joint distributions (more on this below)

11.3 Example: Spam detection

Imagine the problem of trying to automatically detect spam e-mail messages. A simple approach to get started is to look only at the "Subject:" headers in the e-mail messages and attempt to recognize spam by checking some simple computable features. The two simple features we will consider are:

- Whether the subject header is entirely capitalized
- Whether the subject header contains the word 'free', either in upper case or lower case

For example, a message with the subject header "NEW MORTGAGE RATE" is likely to be spam. Simlarly, for "Money for Free", "FREE lunch", etc.

So our model is based on the following three random variables, *Caps*, *Free* and *Spam*, each of which take on the values Y (for Yes) or N (for No)

Caps	=	Y if and only if the subject of the message does not contain lowercase letters
Free	=	Y if and only if the word 'free' appears in the subject (letter case is ignored)
Spam	=	Y if and only if the message is spam

To learn what happens in the real world, one could go into their mailbox, select 100 random messages and count how many times each type of message appears. We might obtain the following table

Free	*Caps*	*Spam*	Number of messages	Estimated probability
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	1.0

The simplest way to estimate the joint distribution from such a sample is just to divide the number of messages of each type by the total number of messages, as shown in the table.

Given a fully specified joint distribution table, we can then lookup the probability of any configuration. For example

$$\begin{aligned} \mathrm{P}(\mathit{Free} = Y, \mathit{Caps} = Y, \mathit{Spam} = Y) &= 0.2 \\ \mathrm{P}(\mathit{Free} = Y, \mathit{Caps} = N, \mathit{Spam} = N) &= 0.0 \\ \text{etc.} \end{aligned}$$

Note that this example shows one drawback of the full joint distribution model: the *sparse data problem*. That is, for any joint distribution there is an exponential number of possible configurations, but any small sample will necessarily only contain a subset of possible patterns. Should one conclude then that every missing pattern has probability zero?

11.4 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 $\theta_{\mathbf{x}^{(\ell)}}$'s, draw a random number p between 0 and 1, and select the configuration $\mathbf{x}^{(\ell)}$ whose cumulative probability interval contains p.

11.5 Evaluation

Evaluate the probability of a complete configuration $\mathbf{x} = (x_1, ..., x_n)$. In the table lookup representation, one can just look up the answer:

$$\mathrm{P}(X_1 = x_1, ..., X_n = x_n) \quad = \quad \theta_{(x_1, x_2, ..., x_n)}$$

11.6 Marginalization

Compute the probability of an *incomplete* configuration.

$$\begin{aligned}
&\mathrm{P}(X_1 = x_1, ..., X_k = x_k) \\
&\quad = \sum_{y_{k+1}} \cdots \sum_{y_n} \mathrm{P}(X_1 = x_1, ..., X_k = x_k, X_{k+1} = y_{k+1}, ..., X_n = y_n) \\
&\quad = \sum_{y_{k+1}} \cdots \sum_{y_n} \theta_{(x_1, ..., x_k, y_{k+1}, ..., y_n)}
\end{aligned}$$

Need to be able to evaluate complete configurations and then sum over m^{n-k} possible completions.

11.7 Conditioning

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

$$\begin{aligned}
&\mathrm{P}(X_{k+1} = y_{k+1}, ..., X_n = y_n | X_1 = x_1, ..., X_k = x_k) \\
&\quad = \frac{\mathrm{P}(X_1 = x_1, ..., X_k = x_k, X_{k+1} = y_{k+1}, ..., X_n = y_n)}{\mathrm{P}(X_1 = x_1, ..., X_k = x_k)} \\
&\quad = \frac{\theta_{(x_1, ..., x_k, y_{k+1}, ..., y_n)}}{\sum_{z_{k+1}} \cdots \sum_{z_n} \theta_{(x_1, ..., x_k, z_{k+1}, ..., z_n)}}
\end{aligned}$$

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

11.8 Completion

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

$$\begin{aligned}
y^*_{k+1}, ..., y^*_n &= \arg\max_{y_{k+1},...,y_n} \mathrm{P}(X_{k+1}{=}y_{k+1}, ..., X_n{=}y_n | X_1{=}x_1, ..., X_k{=}x_k) \\
&= \arg\max_{y_{k+1},...,y_n} \frac{\mathrm{P}(X_1{=}x_1, ..., X_k{=}x_k, X_{k+1}{=}y_{k+1}, ..., X_n{=}y_n)}{\mathrm{P}(X_1{=}x_1, ..., X_k{=}x_k)} \\
&= \arg\max_{y_{k+1},...,y_n} \mathrm{P}(X_1{=}x_1, ..., X_k{=}x_k, X_{k+1}{=}y_{k+1}, ..., X_n{=}y_n) \\
&= \arg\max_{y_{k+1},...,y_n} \theta_{(x_1,...,x_k,y_{k+1},...,y_n)}
\end{aligned}$$

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

11.9 *Structured* probability models

Structured probability models are ways of specifying specialized joint distributions which permit

- more compact representations
- more efficient computation

That is, we will impose structure on joint distribution $\mathrm{P}(X_1{=}x_1, ..., X_n{=}x_n)$. One key tool for imposing structure is variable independence.

Definition Random variables X_1 and X_2 are *independent* if $\mathrm{P}(X_1{=}x_1, X_2{=}x_2) = \mathrm{P}(X_1{=}x_1)\mathrm{P}(X_2{=}x_2)$ for all x_1, x_2. Equivalently, if $\mathrm{P}(X_1{=}x_1|X_2{=}x_2) = \mathrm{P}(X_1{=}x_1)$ for all x_1, x_2.

Definition Random variables X_1 and X_2 are *conditionally independent given* X_3 if $\mathrm{P}(X_1 = x_1, X_2 = x_2|X_3 = x_3) = \mathrm{P}(X_1 = x_1|X_3 = x_3)\mathrm{P}(X_2 = x_2|X_3{=}x_3)$ for all x_1, x_2, x_3. Equivalently, if $\mathrm{P}(X_1{=}x_1|X_2{=}x_2, X_3{=}x_3) = \mathrm{P}(X_1{=}x_1|X_3{=}x_3)$ for all x_1, x_2, x_3.

11.10 Example: Fully independent model

Assume $\mathrm{P}(X_1 = x_1, ..., X_n = x_n) = \mathrm{P}(X_1 = x_1) \cdots \mathrm{P}(X_n = x_n)$. This yields a very restricted form of joint distribution where we can represent each component distribution separately. For a random variable X_j, one can represent $\mathrm{P}(X_j = x)$ by a lookup table with m parameters (minus one constraint). Let $\theta_{j,x}$ denote the probability X_j takes on value x. That is, $\theta_{j,x} = \mathrm{P}(X_j = x)$. These numbers are positive and satisfy the constraint $\sum_{x=1}^{m} \theta_{j,x} = 1$ for each j. Thus, the joint distribution over $X_1, ..., X_n$ 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).

11.10.1 Spam detection example again

The fully independent model is basically useless in our spam detection example because it assumes that the three random variables, *Caps, Free,* and *Spam* are completely independent. That is, knowing whether a subject header is capitalized or contains the word 'free' is assumed to be independent of whether the message is spam. Nevetheless, let us consider what happens in this simple case. (We will consider a more useful model later.) From the previous sample data, one could naively estimate each probability independently:

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

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

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

For example, in this fully independent probability model, the probability that any message is spam is 0.47, regardless of what the values of *Caps* and *Free* are. The probability for a specific configuration, like $\langle \mathit{Caps} = Y, \mathit{Free} = N, \mathit{Spam} = N \rangle$ for example, would now be given by

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

11.10.2 Simulation

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

11.10.3 Evaluation

Given a complete configuration $\mathbf{x} = (x_1, ..., x_n)$, look up the probability of each component x_j and then multiply the answers together.

$$\mathrm{P}(X_1 = x_1, ..., X_n = x_n) \quad = \quad \theta_{1,x_1} \times \cdots \times \theta_{n,x_n}$$

11.10.4 Marginalization

$$\begin{aligned}
&\mathrm{P}(X_1 = x_1, ..., X_k = x_k) \\
&\quad = \sum_{y_{k+1}} \cdots \sum_{y_n} \mathrm{P}(X_1 = x_1, ..., X_k = x_k, X_{k+1} = y_{k+1}, ..., X_n = y_n) \\
&\quad = \sum_{y_{k+1}} \cdots \sum_{y_n} \mathrm{P}(X_1 = x_1) \cdots \mathrm{P}(X_k = x_k) \mathrm{P}(X_{k+1} = y_{k+1}) \cdots \mathrm{P}(X_n = y_n) \\
&\quad = \mathrm{P}(X_1 = x_1) \cdots \mathrm{P}(X_k = x_k) \left[\sum_{y_{k+1}} \mathrm{P}(X_{k+1} = y_{k+1}) \left[\sum_{y_{k+2}} \cdots \left[\sum_{y_n} \mathrm{P}(X_n = y_n) \right] \right] \right] \\
&\quad = \mathrm{P}(X_1 = x_1) \cdots \mathrm{P}(X_k = x_k) \left[\sum_{y_{k+1}} \mathrm{P}(X_{k+1} = y_{k+1}) \right] \cdots \left[\sum_{y_n} \mathrm{P}(X_n = y_n) \right] \\
&\quad = \mathrm{P}(X_1 = x_1) \cdots \mathrm{P}(X_k = x_k) \\
&\quad = \theta_{1,x_1} \times \cdots \times \theta_{k,x_k}
\end{aligned}$$

Only have to lookup and multiply k numbers.

11.10.5 Conditioning

$$
\begin{aligned}
&\mathrm{P}(X_{k+1}{=}y_{k+1}, ..., X_n{=}y_n | X_1{=}x_1, ..., X_k{=}x_k) \\
&= \frac{\mathrm{P}(X_1{=}x_1, ..., X_k{=}x_k, X_{k+1}{=}y_{k+1}, ..., X_n{=}y_n)}{\mathrm{P}(X_1{=}x_1, \ldots, X_k{=}x_k)} \\
&= \frac{\mathrm{P}(X_1{=}x_1)\cdots\mathrm{P}(X_k{=}x_k)\mathrm{P}(X_{k+1}{=}y_{k+1})\cdots\mathrm{P}(X_n{=}y_n)}{\mathrm{P}(X_1{=}x_1)\cdots\mathrm{P}(X_k{=}x_k)} \\
&= \mathrm{P}(X_{k+1}{=}y_{k+1})\cdots\mathrm{P}(X_n{=}y_n) \\
&= \theta_{k+1,y_{k+1}} \times \cdots \times \theta_{n,y_n}
\end{aligned}
$$

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

11.10.6 Completion

$$
\begin{aligned}
y^*_{k+1}, ..., y^*_n &= \arg\max_{y_{k+1},\ldots,y_n} \mathrm{P}(X_{k+1}{=}y_{k+1}, ..., X_n{=}y_n | X_1{=}x_1, ..., X_k{=}x_k) \\
&= \arg\max_{y_{k+1},\ldots,y_n} \mathrm{P}(X_{k+1}{=}y_{k+1})\cdots\mathrm{P}(X_n{=}y_n) \\
&= \arg\left[\max_{y_{k+1}} \mathrm{P}(X_{k+1}{=}y_{k+1}) \left[\max_{y_{k+2}} \cdots \left[\max_{y_n} \mathrm{P}(X_n{=}y_n)\right]\right]\right] \\
&\quad \text{(Since max distributes over product just like sum. That is,} \\
&\quad \max_i ax_i = a\max_i x_i \text{ (for } a, x_i \geq 0\text{) just like } \textstyle\sum_i ax_i = a\sum_i x_i.\text{)} \\
&= \arg\left[\max_{y_{k+1}} \mathrm{P}(X_{k+1}{=}y_{k+1})\right]\cdots\left[\max_{y_n} \mathrm{P}(X_n{=}y_n)\right] \\
&= \arg\left[\max_{y_{k+1}} \theta_{k+1,y_{k+1}}\right]\cdots\left[\max_{y_n} \theta_{n,y_n}\right]
\end{aligned}
$$

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

11.10.7 Note: Distributive/associative laws

It is important to note a general rule which we used to separate summations in the above calculations: 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 sum's and max's distribute over products in the following

identical way.

$$
\begin{aligned}
\sum_a \sum_b f(a)g(b) &= \sum_a f(a) \left(\sum_b g(b) \right) \\
&\quad \text{since } f(a) \text{ constant when summing over } b \\
&= \left(\sum_b g(b) \right) \left(\sum_a f(a) \right) \\
&\quad \text{since } \sum_b g(b) \text{ constant when summing over } a \\
&= \left(\sum_a f(a) \right) \left(\sum_b g(b) \right)
\end{aligned}
$$

If we assume that $f(a) \geq 0$ and $g(b) \geq 0$, then the same rules apply for max:

$$
\begin{aligned}
\max_a \max_b f(a)g(b) = \; &= \max_a f(a) \left(\max_b g(b) \right) \\
&\quad \text{since } f(a) \text{ constant when maximizing over } b \\
&= \left(\max_b g(b) \right) \left(\max_a f(a) \right) \\
&\quad \text{since } \max_b g(b) \text{ constant when maximizing over } a \\
&= \left(\max_a f(a) \right) \left(\max_b g(b) \right)
\end{aligned}
$$

Readings

Russell and Norvig: Chap 14
Dean, Allen, Aloimonos: Sect 8.2