Algorithms for Graphical Models (AGM)

Factors

$Date: 2007/10/10 15:07:37 $

AGM-05

In this lecture

- Factors
- Factor algebra
  - Marginalisation
  - Multiplication
- Independence models

Recall: A joint probability distribution

Bronchitis	Cancer	Smoker	
absent	absent	nonsmoker	0.030000
absent	absent	smoker	0.000000
absent	present	nonsmoker	0.270000
absent	present	smoker	0.150000
present	absent	nonsmoker	0.020000
present	absent	smoker	0.000000
present	present	nonsmoker	0.180000
present	present	smoker	0.350000

Factors

- This joint probability distribution is represented by a *factor*.

- A factor is simply a mapping from joint instantiations of variables to real numbers.

- Factors are the workhorses of graphical models.

- Most joint probability distributions we will study are defined using more than one factor. Saturated models are thus unusual.

Factors formally (almost)

- Let Δ be a set of variables.

- In our example $\Delta = \{\text{Bronchitis}, \text{Cancer}, \text{Smoker}\}$.

- For each variable $\delta \in \Delta$, define $\mathcal{I}_\delta$ to be the variable's values.

- In our example we have, $\mathcal{I}_{\text{Bronchitis}} = \{\text{absent}, \text{present}\}$, $\mathcal{I}_{\text{Cancer}} = \{\text{absent}, \text{present}\}$, $\mathcal{I}_{\text{Smoker}} = \{\text{nonsmoker}, \text{smoker}\}$

Factors formally (almost) (ctd)

- Let $\mathcal{I} = \times_{\delta \in \Delta} \mathcal{I}_\delta$ be called a *table*.

- In our example $\mathcal{I} = \{\text{absent}, \text{present}\} \times \{\text{absent}, \text{present}\} \times \{\text{nonsmoker}, \text{smoker}\}$

- Each element i of $\mathcal{I}$ is a vector of values and is called a *cell*.

- In our example, $(\text{absent}, \text{absent}, \text{smoker})$ and $(\text{absent}, \text{present}, \text{nonsmoker})$ are both cells.

- A factor with table $\mathcal{I}$ is simply a function $f : \mathcal{I} \rightarrow \mathcal{R}$.

Marginalisation

- Marginalisation 'throws some information away' from a factor by reducing the number of variables involved.

- Easiest to understand by looking at contingency tables (which are also factors).

- Cue `marginalise_demo` from `gPy.Examples`

Marginalisation = summing out

- Each cell in the marginal table is simply the sum of all the ‘corresponding’ cells in the original table.

- Think of marginalisation as ‘projecting down’ a factor to a smaller number of dimensions

Marginalisation formally

- Let $f(i)$ be the value associated with cell i in some factor with variables Δ and table $\mathcal{I}$.

- Let $a \subset \Delta$, then $\mathcal{I}_a$, the table for the marginal factor defined by a is given by $\mathcal{I}_a = \times_{\delta \in a} \mathcal{I}_\delta$.

- For any cell $i \in \mathcal{I}$, let i_δ be its value for variable δ.

- Let i_a be the projection of i onto $a \subset \Delta$ defined as: $\times_{\delta \in a} i_\delta$

- $f_a : \mathcal{I}_a \rightarrow \mathcal{R}$, the function defining the marginal factor is defined as follows: $f_a(i_a) = \sum_{j \in \mathcal{I}: j_a = i_a} f(j)$

Marginalisation informally

- Suppose factor f involves variables A, B, C, D (i.e. these variables define its table). It is normal to reflect this by writing f as $f(A, B, C, D)$.

- This is informal since the factor is actually a function operating on tuples of *values* of its variables, not on the variables themselves.

With this informal notation, marginalising away A would be written:

$$f'(B, C, D) = \sum_A f(A, B, C, D)$$

Marginalisation for probability distributions

- Recall that if 'James-in-York' and 'James-in-Oxford' are mutually exclusive events then $P(\text{James-in-York} \cup \text{James-in-Oxford}) = P(\text{James-in-York}) + P(\text{James-in-Oxford})$, where 'James-in-York $\cup$ James-in-Oxford' is the event that one of James-in-York or James-in-Oxford happens.

- A joint probability distribution (like all probability distributions) assigns probabilities to a *sample space* of events.

- Each joint instantiation is an event, *and they are all mutually exclusive*.

Marginal distributions

- A *marginal distribution* is one computed by summing out one or more variables from some joint distribution.

- If the original joint distribution is defined by a single factor, then it is computed by marginalising that factor.

- Cue `marginalise_.demo2` from `gPy.Examples`

Events/cells in the marginal table

$$
\begin{aligned}
&(Bronchitis = absent) \\
&\quad = \quad (Bronchitis = absent, Cancer = absent, Smoking = nonsmoker) \\
&\quad \vee \quad (Bronchitis = absent, Cancer = absent, Smoking = smoker) \\
&\quad \vee \quad (Bronchitis = absent, Cancer = present, Smoking = nonsmoker) \\
&\quad \vee \quad (Bronchitis = absent, Cancer = present, Smoking = smoker)
\end{aligned}
$$

Since exclusive, we can add probabilities: $0.03 + 0.0 + 0.27 + 0.15 = 0.45$

On the importance of marginal distributions

- Big joint probability distributions are difficult to make sense of, hence the need to compute less informative marginal distributions.

- Usually we want marginal distributions with a single variable.

- Many of the algorithms we will study are efficient ways to compute marginal distributions for all variables in a joint distribution.

Independence

$A = a$ and $B = b$ are two *independent events* iff $P(A = a \cap B = b) = P(A = a)P(B = b)$.

Two random variables C and D are independent in a joint distribution P iff

$$\forall c \in \mathcal{I}_C, d \in \mathcal{I}_D : P(C = c, D = d) = P(C = c)P(D = d)$$

where $P(C = c) = \sum_{d \in \mathcal{I}_D} P(C = c, D = d)$ is the marginal probability of $C = c$, and $P(D = d)$ is the marginal probability of $D = d$.

NB. This is a strong condition: independence has to hold for *all* combinations of values.

Factor multiplication

- If C and D are independent and the distributions for C and D are each represented by a single factor, [denoted $P(C)$ and $P(D)$, resp.] then computing a new factor for the joint [denoted $P(C, D)$] is just a question of multiplying the appropriate values.

- To do this we 'broadcast' each factor so that both have the same variables, and then apply pointwise multiplication.

- Cue `prod_demo` from `gPy.Examples`

Factors formally (really!)

- Each factor involved in defining a joint distribution—for example $P(\text{Cancer}, \text{VisitAsia})$—can be defined on *the same table*.

- In our example, we can have some table $\mathcal{I}$ such that
  $P(\text{Cancer}) : \mathcal{I} \rightarrow \mathcal{R}$ and
  $P(\text{VisitAsia}) : \mathcal{I} \rightarrow \mathcal{R}$.

- This means that factor multiplication is normal pointwise function multiplication.

- This table involves all variables in the joint distribution.

Factors formally (really!) (ctd.)

- However, particular factors typically only depend on a subset of all the variables in the joint distribution. So we take advantage of this to give them a compact representation.

- The data broadcasting makes the existence of this shared table more obvious.

Independence and factored models

- If two random variables C and D are independent then their joint distribution $P(C, D)$ is simply the product of the two relevant marginal distributions $P(C)$ and $P(D)$.

- It follows that the joint distribution can be represented by these two factors.

- This is our first example of a *factored model*.

- Similarly, if we have n mutually independent random variables, then their joint distribution can be represented by n factors.

Independence model

- Recall that a model is a set of probability distributions.

- For a set of n variables, let the *independence model* be the set of all distributions where there is independence between all variables.

- Any distribution in this model can be represented by n univariate distributions (each represented by a factor).

The blessings of independence

- Suppose each variable has m values, then if no independence relations exist (the saturated model), a single factor with m^n numbers is needed.

- Assuming the independence model, n factors each with m values are needed: only nm numbers in total.

- (You can save a bit more if you're desperate since probabilities have to sum to one.)

- Computations are also far cheaper with the independence model

From independence to conditional independence

- The big problem is that such complete independence rarely models the world well.

- We need models somewhere in between the saturated and independence models.

- And so to models which use *conditional independence* . . .