MCGs: Transitions and Games

Oliver Richardson `oli@cs.cornell.edu`

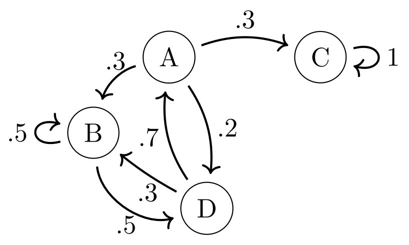
September 2, 2019

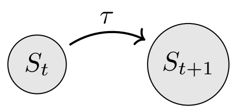
There are two different styles of common graphical models, and though their relation is quite clear, they are related in important ways, which is not often enough mentioned explicitly.

My marginal constraint graphs can capture both.

The first class of graphical model is BN-like: they represent factorizations of probability distributions. The interpretation

The other class of model is automaton-like: a transition model between states.

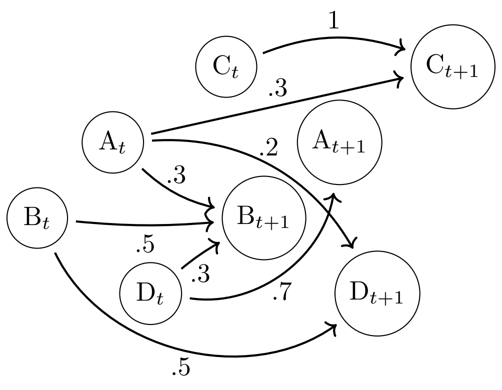




The two pictures represent the same transition matrix:

$$\tau = \begin{array}{c c} \begin{matrix} \text{A} & \text{B} & \text{C} & \text{D} \end{matrix} & \\ \begin{bmatrix} 0 & 0.3 & 0.3 & 0.2 \\ 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 1 & 0 \\ 0.7 & 0.3 & 0 & 0 \end{bmatrix} & \begin{matrix} \text{A} \\ \text{B} \\ \text{C} \\ \text{D} \end{matrix} \end{array}$$

but both hide something. On the left we see the individual states, but we've projected out across all of time; on the right, we explicitly see time, but have abstracted out all of the states. We could also have considered a different, cluttered representation which shows both at once, tagging both the time and the states:



At the other extreme, we could have compressed in both directions, giving us

$$\begin{tikzcd} S \arrow[loop above, "\tau"] \end{tikzcd}$$

Two Interpretations

As usual, there are multiple ways of interpreting the probability:

1. The specification τ is somehow actually the truth about what happens to S over times. It is defined as the random process which has the statistics in question.
2. τ is just fit to experiential data — a best case approximation, marginalizing out the things we weren't able (or didn't care to) control for.

While both

Temporal. It is common to model transitions using automata-like graphical models

1 Encoding as MCGs

Structurally automaton-like graphs are just a special case making heavy use of our extra element $\bullet$ and the fact that we don't require conditioning on it. Each element $s \in S$ can be encoded as a node $\mathbb{1}_s$

There is, of course, also an explicit conversion between the two of them, involving packing and unpacking. If the BN-like model is a graph $(V, E, \mathcal{S}, \mathbf{p})$, then we can create a new model whose nodes are every possible value of any node, and whose edges are the unpacked fibers of the edges in the original graph, i.e.,

$$\left(\bigcup_{v:V} \{ \mathbb{1}_x : x \in \mathcal{S}_v \}, \quad \bigcup_{(A,B):E} (\mathcal{S}_A \times \mathcal{S}_B) = \Big\{ (a_i, b_j) \in \mathcal{S}_A \times \mathcal{S}_B : a_i, b_i \Big\} \right),$$

if we just

[1]

2 Relation to Preferences

Any binary relation can be turned into a transition graph by normalizing, which gives you the maximum entropy transition graph consistent with a

We can also make use of constraints in MCG's to specify fixed points.

3 Game Trees

[1] This unpacking / re-packing is in fact exactly the Category of Elements construction