1 Why embedded conditioning matters

1.1 Computing the likelihood for the conditioned variable

Joint distribution on \((x, y)\)
\[
p_1 = \text{dist bern 0.5}
\]
\[
y = \text{if } x \text{ then condition 10 norm 11 else condition 10 norm 11}
\]
\[
\text{return } x
\]

Conditioning on \(y\) being \(10\)
\[
p_{1c} = \text{dist bern 0.5}
\]
\[
\text{observe } (y = 10) \quad \text{-- Not a valid Hakaru10 statement!}
\]
\[
\text{return } x
\]

Valid conditioning on \(y\) being \(10\)
\[
p_{1c} = \text{dist bern 0.5}
\]
\[
\text{observe } (y = 10) \quad \text{where to get } y\text{'s likelihood from?}
\]
\[
\text{return } x
\]

A better idea is to push the conditioning statement into \(p_1\), where \(y\) is computed and where its distribution is obvious. Hence we get \(p_{1c}\).

1.2 Modularity

If we are allowed to compose previously written models into new ones, what used to be ‘top level’ conditioning quickly ends up buried inside.

2 Warm-up: Models with branching

Deriving the correct Wingate et al. formula
\[
p_2 = \text{do } x \leftarrow \text{dist bern a}
\]
\[
y = \text{if } x \text{ then do } (y t \leftarrow \text{et; return } y t) \text{ else do } (y f \leftarrow \text{et; return } y f)
\]
\[
\text{return } (x, y)
\]

Acceptance ratio for \(s_1 \rightarrow s_2\)
\[
\frac{\alpha(s_1, s_2)}{\gamma(s_1, s_2)} = \text{min}[1, r(s_2, s_1)]
\]
\[
r(s_2, s_1) = \frac{\pi(s_2) p_2(c)}{\pi(s_1) p_2(c')}
\]

Proposal kernel \(\pi(s)\); \(\pi(s)\) target density

2.1 Wingate-like method

- A model (= program): a DAG of elementary random primitives (ERP) like \texttt{bern} in \(p_2\)
- Each ERP is uniquely named (contra-Wingate)
- \(p\): the number of active ERPs in \(p\)
- A sample from a program (= trace): a set of samples of all ERPs
- To sample from a program, we build a Markov Chain over the space of traces, by proposing an update to one ERP sample

2.2 MH sampling from \(p_2\)

Current state \(s_1 = (x = \text{true}, y = yf)\)
\[
\pi(s_1) = \pi(x = \text{true}) \delta(y = yf) \pi(c = c') \pi(c = c')
\]

Proposed state \(s_2 = (x = \text{false}, y = yf)\)
\[
\pi(s_2) \text{ similarly}
\]

Proposal kernel \(q(s_1, s_2)\)
- \(x\) from all other eligible ERPs: \(1/|s|\)
- \(x\) from \(x\) from \(x\) to \(x\) to \(x\); \(h_1\)
- switch from \(y = yf\) to \(y = yf\):
- \(r(s_2, s_1) = \frac{(1 - a) (b/yf/zf) \cdot (1 + e)(1 + e) (f)}{1 + e f (1 + e f)}
\]

which is the correct (in the 3d revision) Wingate formula

3 Models with conditioning and branching

\[
p_{1c} = \text{do } x \leftarrow \text{dist bern a}
\]
\[
y = \text{if } x \text{ then do } (y t \leftarrow \text{et; return } y t) \text{ else do } (y f \leftarrow \text{et; return } y f)
\]
\[
\text{return } (x, y)
\]

To be explicit
\[
p_3 = \text{do } x \leftarrow \text{dist bern a}
\]
\[
y = \text{if } x \text{ then do } (y t \leftarrow \text{et; return } y t)
\]
\[
\text{else do } (y f \leftarrow \text{et; return } y f)
\]
\[
\text{return } (x, y)
\]

\[
p_{1c} = \text{p3 conditioned on } z = 0
\]

Consider the same transition as before \(s_1 = (x = \text{true}, y = yf) \rightarrow s_2 = (x = \text{false}, y = yf)\)
\[
\pi(s_1) = \pi(x = \text{true}) \delta(y = yf) \pi(c = c') \pi(c = c')
\]

Additional factor \(\pi(z = 0)\)

Acceptance ratio
\[
r(s_2, s_1) = \frac{(1 - a) (b/yf/zf) \cdot (1 + e)(1 + e) (f)}{1 + e f (1 + e f)}
\]

Additional factor scoring the observation \(z = 0\) within the distributions of \(x\) and \(y\).

4 Implementation

Hakaru10 http://okmij.org/ftp/kakuritu/Hakaru10/