always reports a posterior probability for the query of $\langle 1.0, 0.0 \rangle$; if it starts in [*false*,*false*] it always reports a posterior probability for the query of $\langle 0.0, 1.0 \rangle$.

Gibbs sampling fails in this case because the deterministic relationship between *Cloudy* and *Rain* breaks the property of ergodicity that is required for convergence. If, however, we make the relationship *nearly* deterministic, then convergence is restored, but happens arbitrarily slowly. There are several fixes that help MCMC algorithms mix more quickly. One is **block sampling**: sampling multiple variables simultaneously. In this case, we could sample *Cloudy* and *Rain* jointly, conditioned on their combined Markov blanket. Another is to generate next states more intelligently, as we will see in the next section.

Block sampling

Metropolis–Hastings sampling

The Metropolis–Hastings or MH sampling method is perhaps the most broadly applicable MCMC algorithm. Like Gibbs sampling, MH is designed to generate samples $\mathbf{x}$ (eventually) according to target probabilities $\pi(\mathbf{x})$; in the case of inference in Bayesian networks, we want $\pi(\mathbf{x}) = P(\mathbf{x} \mid \mathbf{e})$. Like simulated annealing (page 115), MH has two stages in each iteration of the sampling process:

1. Sample a new state $\mathbf{x}'$ from a **proposal distribution** $q(\mathbf{x}' \mid \mathbf{x})$, given the current state $\mathbf{x}$.
2. Probabilistically accept or reject $\mathbf{x}'$ according to the **acceptance probability**

Proposal distribution

Acceptance probability

$$a(\mathbf{x}' \mid \mathbf{x}) = \min\left(1, \frac{\pi(\mathbf{x}')q(\mathbf{x} \mid \mathbf{x}')}{\pi(\mathbf{x})q(\mathbf{x}' \mid \mathbf{x})}\right).$$

If the proposal is rejected, the state remains at $\mathbf{x}$.

The transition kernel for MH consists of this two-step process. Note that if the proposal is rejected, the chain stays in the same state.

The proposal distribution is responsible, as its name suggests, for proposing a next state $\mathbf{x}'$. For example, $q(\mathbf{x}' \mid \mathbf{x})$ could be defined as follows:

- With probability 0.95, perform a Gibbs sampling step to generate $\mathbf{x}'$.
- Otherwise, generate $\mathbf{x}'$ by running the WEIGHTED-SAMPLE algorithm from page 440.

This proposal distribution causes MH to do about 20 steps of Gibbs sampling then "restarts" the process from a new state (assuming it is accepted) that is generated from scratch. By this stratagem, it gets around the problem of Gibbs sampling getting stuck in one part of the state space and being unable to reach the other parts.

You might ask how on Earth we know that MH with such a weird proposal actually converges to the right answer. The remarkable thing about MH is that *convergence to the correct stationary distribution is guaranteed for* any *proposal distribution,* provided the resulting transition kernel is ergodic.

This property follows from the way the acceptance probability is defined. As with Gibbs sampling, the self-loop with $\mathbf{x} = \mathbf{x}'$ automatically satisfies detailed balance, so we focus on the case where $\mathbf{x} \neq \mathbf{x}'$. This can occur only if the proposal is accepted. The probability of such a transition occurring is

$$k(\mathbf{x} \to \mathbf{x}') = q(\mathbf{x}' \mid \mathbf{x})a(\mathbf{x}' \mid \mathbf{x}).$$

As with Gibbs sampling, proving detailed balance means showing that the flow from $\mathbf{x}$ to $\mathbf{x}'$, $\pi(\mathbf{x})k(\mathbf{x} \to \mathbf{x}')$, matches the flow from $\mathbf{x}'$ to $\mathbf{x}$, $\pi(\mathbf{x}')k(\mathbf{x}' \to \mathbf{x})$. After plugging in the