

process and resample all n variables using the updated values of their neighbors. Asymptotically, after many repetitions, this process converges to sampling from the correct distribution. It can be difficult to determine when the samples have reached a sufficiently accurate approximation of the desired distribution. Sampling techniques for undirected models are an advanced topic, covered in more detail in chapter 17.

16.4 Advantages of Structured Modeling

The primary advantage of using structured probabilistic models is that they allow us to dramatically reduce the cost of representing probability distributions as well as learning and inference. Sampling is also accelerated in the case of directed models, while the situation can be complicated with undirected models. The primary mechanism that allows all of these operations to use less runtime and memory is choosing to not model certain interactions. Graphical models convey information by leaving edges out. Anywhere there is not an edge, the model specifies the assumption that we do not need to model a direct interaction.

A less quantifiable benefit of using structured probabilistic models is that they allow us to explicitly separate representation of knowledge from learning of knowledge or inference given existing knowledge. This makes our models easier to develop and debug. We can design, analyze, and evaluate learning algorithms and inference algorithms that are applicable to broad classes of graphs. Independently, we can design models that capture the relationships we believe are important in our data. We can then combine these different algorithms and structures and obtain a Cartesian product of different possibilities. It would be much more difficult to design end-to-end algorithms for every possible situation.

16.5 Learning about Dependencies

A good generative model needs to accurately capture the distribution over the observed or “visible” variables \mathbf{v} . Often the different elements of \mathbf{v} are highly dependent on each other. In the context of deep learning, the approach most commonly used to model these dependencies is to introduce several latent or “hidden” variables, \mathbf{h} . The model can then capture dependencies between any pair of variables v_i and v_j indirectly, via direct dependencies between v_i and \mathbf{h} , and direct dependencies between \mathbf{h} and v_j .

A good model of \mathbf{v} which did not contain any latent variables would need to