

have very large numbers of parents per node in a Bayesian network or very large cliques in a Markov network. Just representing these higher order interactions is costly—both in a computational sense, because the number of parameters that must be stored in memory scales exponentially with the number of members in a clique, but also in a statistical sense, because this exponential number of parameters requires a wealth of data to estimate accurately.

When the model is intended to capture dependencies between visible variables with direct connections, it is usually infeasible to connect all variables, so the graph must be designed to connect those variables that are tightly coupled and omit edges between other variables. An entire field of machine learning called **structure learning** is devoted to this problem. For a good reference on structure learning, see (Koller and Friedman, 2009). Most structure learning techniques are a form of greedy search. A structure is proposed, a model with that structure is trained, then given a score. The score rewards high training set accuracy and penalizes model complexity. Candidate structures with a small number of edges added or removed are then proposed as the next step of the search. The search proceeds to a new structure that is expected to increase the score.

Using latent variables instead of adaptive structure avoids the need to perform discrete searches and multiple rounds of training. A fixed structure over visible and hidden variables can use direct interactions between visible and hidden units to impose indirect interactions between visible units. Using simple parameter learning techniques we can learn a model with a fixed structure that imputes the right structure on the marginal $p(\mathbf{v})$.

Latent variables have advantages beyond their role in efficiently capturing $p(\mathbf{v})$. The new variables \mathbf{h} also provide an alternative representation for \mathbf{v} . For example, as discussed in section 3.9.6, the mixture of Gaussians model learns a latent variable that corresponds to which category of examples the input was drawn from. This means that the latent variable in a mixture of Gaussians model can be used to do classification. In chapter 14 we saw how simple probabilistic models like sparse coding learn latent variables that can be used as input features for a classifier, or as coordinates along a manifold. Other models can be used in this same way, but deeper models and models with different kinds of interactions can create even richer descriptions of the input. Many approaches accomplish feature learning by learning latent variables. Often, given some model of \mathbf{v} and \mathbf{h} , experimental observations show that $\mathbb{E}[\mathbf{h} \mid \mathbf{v}]$ or $\operatorname{argmax}_{\mathbf{h}} p(\mathbf{h}, \mathbf{v})$ is a good feature mapping for \mathbf{v} .