19.4 Examples of MRFs

In this section, we show how several popular probability models can be conveniently expressed as UGMs.

19.4.1 Ising model

The **Ising model** is an example of an MRF that arose from statistical physics.[3] It was originally used for modeling the behavior of magnets. In particular, let $y_s \in \{-1, +1\}$ represent the spin of an atom, which can either be spin down or up. In some magnets, called **ferro-magnets**, neighboring spins tend to line up in the same direction, whereas in other kinds of magnets, called **anti-ferromagnets**, the spins "want" to be different from their neighbors.

We can model this as an MRF as follows. We create a graph in the form of a 2D or 3D lattice, and connect neighboring variables, as in Figure 19.1(b). We then define the following pairwise clique potential:

$$\psi_{st}(y_s, y_t) = \begin{pmatrix} e^{w_{st}} & e^{-w_{st}} \\ e^{-w_{st}} & e^{w_{st}} \end{pmatrix} \tag{19.17}$$

Here w_{st} is the coupling strength between nodes s and t. If two nodes are not connected in the graph, we set $w_{st} = 0$. We assume that the weight matrix $\mathbf{W}$ is symmetric, so $w_{st} = w_{ts}$. Often we assume all edges have the same strength, so $w_{st} = J$ (assuming $w_{st} \neq 0$).

If all the weights are positive, $J > 0$, then neighboring spins are likely to be in the same state; this can be used to model ferromagnets, and is an example of an **associative Markov network**. If the weights are sufficiently strong, the corresponding probability distribution will have two modes, corresponding to the all +1's state and the all -1's state. These are called the **ground states** of the system.

If all of the weights are negative, $J < 0$, then the spins want to be different from their neighbors; this can be used to model an anti-ferromagnet, and results in a **frustrated system**, in which not all the constraints can be satisfied at the same time. The corresponding probability distribution will have multiple modes. Interestingly, computing the partition function $Z(J)$ can be done in polynomial time for associative Markov networks, but is NP-hard in general (Cipra 2000).

There is an interesting analogy between Ising models and Gaussian graphical models. First, assuming $y_t \in \{-1, +1\}$, we can write the unnormalized log probability of an Ising model as follows:

$$\log \tilde{p}(\mathbf{y}) = -\sum_{s \sim t} y_s w_{st} y_t = -\frac{1}{2}\mathbf{y}^T \mathbf{W} \mathbf{y} \tag{19.18}$$

(The factor of $\frac{1}{2}$ arises because we sum each edge twice.) If $w_{st} = J > 0$, we get a low energy (and hence high probability) if neighboring states agree.

Sometimes there is an **external field**, which is an energy term which is added to each spin. This can be modelled using a local energy term of the form $-\mathbf{b}^T \mathbf{y}$, where $\mathbf{b}$ is sometimes called

[3] Ernst Ising was a German-American physicist, 1900–1998.

a **bias term**. The modified distribution is given by

$$\log \tilde{p}(\mathbf{y}) \quad = \quad \sum_{s\sim t} w_{st} y_s y_t + \sum_s b_s y_s = \frac{1}{2}\mathbf{y}^T\mathbf{W}\mathbf{y} + \mathbf{b}^T\mathbf{y} \tag{19.19}$$

where $\boldsymbol{\theta} = (\mathbf{W}, \mathbf{b})$.

If we define $\boldsymbol{\mu} \triangleq -\frac{1}{2}\boldsymbol{\Sigma}^{-1}\mathbf{b}$, $\boldsymbol{\Sigma}^{-1} = -\mathbf{W}$, and $c \triangleq \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$, we can rewrite this in a form that looks similar to a Gaussian:

$$\tilde{p}(\mathbf{y}) \propto \exp(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\boldsymbol{\mu}) + c) \tag{19.20}$$

One very important difference is that, in the case of Gaussians, the normalization constant, $Z = |2\pi\boldsymbol{\Sigma}|$, requires the computation of a matrix determinant, which can be computed in $O(D^3)$ time, whereas in the case of the Ising model, the normalization constant requires summing over all 2^D bit vectors; this is equivalent to computing the matrix permanent, which is NP-hard in general (Jerrum et al. 2004).

19.4.2 Hopfield networks

A **Hopfield network** (Hopfield 1982) is a fully connected Ising model with a symmetric weight matrix, $\mathbf{W} = \mathbf{W}^T$. These weights, plus the bias terms $\mathbf{b}$, can be learned from training data using (approximate) maximum likelihood, as described in Section 19.5.[4]

The main application of Hopfield networks is as an **associative memory** or **content addressable memory**. The idea is this: suppose we train on a set of fully observed bit vectors, corresponding to patterns we want to memorize. Then, at test time, we present a partial pattern to the network. We would like to estimate the missing variables; this is called **pattern completion**. See Figure 19.7 for an example. This can be thought of as retrieving an example from memory based on a piece of the example itself, hence the term "associative memory".

Since exact inference is intractable in this model, it is standard to use a coordinate descent algorithm known as **iterative conditional modes** (ICM), which just sets each node to its most likely (lowest energy) state, given all its neighbors. The full conditional can be shown to be

$$p(y_s = 1|\mathbf{y}_{-s}, \boldsymbol{\theta}) = \text{sigm}(\mathbf{w}_{s,:}^T\mathbf{y}_{-s} + b_s) \tag{19.21}$$

Picking the most probable state amounts to using the rule $y_s^* = 1$ if $\sum_t w_{st} y_t > b_s$ and using $y_s^* = 0$ otherwise. (Much better inference algorithms will be discussed later in this book.)

Since inference is deterministic, it is also possible to interpret this model as a **recurrent neural network**. (This is quite different from the feedforward neural nets studied in Section 16.5; they are univariate conditional density models of the form $p(y|\mathbf{x}, \boldsymbol{\theta})$ which can only be used for supervised learning.) See Hertz et al. (1991) for further details on Hopfield networks.

A **Boltzmann machine** generalizes the Hopfield / Ising model by including some hidden nodes, which makes the model representationally more powerful. Inference in such models often uses Gibbs sampling, which is a stochastic version of ICM (see Section 24.2 for details).

[4] ML estimation works much better than the outer product rule proposed in in (Hopfield 1982), because it not only lowers the energy of the observed patterns, but it also raises the energy of the non-observed patterns, in order to make the distribution sum to one (Hillar et al. 2012).

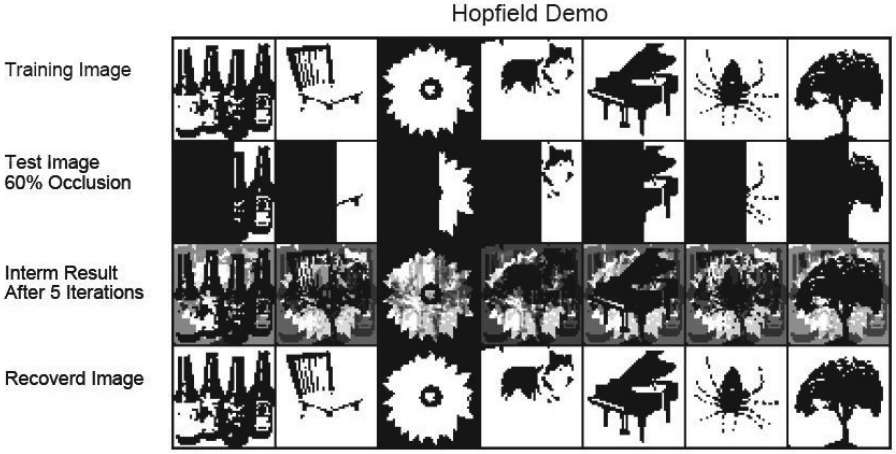

Figure 19.7 Examples of how an associative memory can reconstruct images. These are binary images of size 50×50 pixels. Top: training images. Row 2: partially visible test images. Row 3: estimate after 5 iterations. Bottom: final state estimate. Based on Figure 2.1 of Hertz et al. (1991). Figure generated by `hopfieldDemo`.

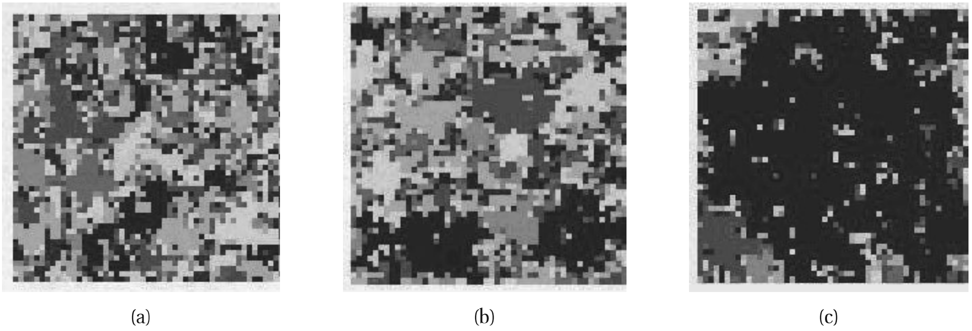

Figure 19.8 Visualizing a sample from a 10-state Potts model of size 128×128 for different association strengths: (a) $J = 1.42$, (b) $J = 1.44$, (c) $J = 1.46$. The regions are labeled according to size: blue is largest, red is smallest. Used with kind permission of Erik Sudderth. See `gibbsDemoIsing` for Matlab code to produce a similar plot for the Ising model.

However, we could equally well apply Gibbs to a Hopfield net and ICM to a Boltzmann machine: the inference algorithm is not part of the model definition. See Section 27.7 for further details on Boltzmann machines.