Exercises

Chapter 10

1. Data file `c10p1.mat` contains 100 data points drawn from the same two-dimensional distribution as those in figure 10.1. Fit a mixture of two circular Gaussian distributions to these data using EM, as in equation 10.4. Do not allow the variance of either of the Gaussians to become smaller than a minimal value of 0.0001.

2. Explore what happens to the fit of the mixture of Gaussians model from exercise 1 as the number of data points from each Gaussian is reduced and the number of potential Gaussians is increased. If you set the minimal variance given in exercise 1 to 0, a Gaussian distribution can settle around a single sample point and then have its variance shrink to 0. Why does this pathological behavior occur?

3. Modify your code from exercise 1 to calculate function $\mathcal{F}$ of equation 10.14 during each E and M step of EM. Check that $\mathcal{F}$ changes monotonically. Explicitly calculate the true log likelihood of the data from equation 10.7 at the end of each M phase. Is it equal to $\mathcal{F}$?

4. Modify the code in exercise 1 to fit a K-means model rather than a mixture of Gaussians. Can you see any practical differences in the solutions that arise?

5. Consider the factor analysis model of figure 10.3 (discussed in more generality later in the chapter). Using the joint probability over v and $\mathbf{u}$ given in equation 10.15, derive an expression for $\mathcal{F}$, and thus the learning rules of equation 10.5.

6. Using the EM version of factor analysis (see the appendix of chapter 10), reproduce figure 10.4. MATLAB® file `c10p6.m` shows how to generate data `u1` for figure 10.4A and B, and `u2` for C and D. First perform factor analysis on these data and reproduce figures 10.4A and C. Next, use the `eig` function to perform principal components analysis on `u1` and `u2`, and thereby produce the rest of the figures. For some initial conditions, the cloud of points in the figures might slope downwards instead of upwards. Why? Calculate the expression for $\mathcal{F}$ derived in exercise 5 as factor analysis progresses and show that it changes monotonically.

7. Apply a rotation matrix to the data set `u2` from exercise 6 (an example rotation matrix is given as `rot` in MATLAB® file `c10p6.m`). Perform factor analysis and principal components analysis on the rotated data. How do the results compare with those for the unrotated data (remember to rotate your results back, if necessary, so that appropriate comparisons can be made)?

8. Construct a data set $\mathbf{u}$ from a set of independent, heavy-tailed, "sources" $\mathbf{v}$ through the relation $\mathbf{u} = \mathbf{G} \cdot \mathbf{v}$. Both $\mathbf{u}$ and $\mathbf{v}$ should

be four-dimensional vectors. Choose the components of $\mathbf{v}$ independently and randomly from a double exponential distribution, for which the probability of getting the value v is proportional to $\exp(-|v|)$ (note that a one-sided exponentially distributed random variable can be generated using either `exprnd(·)` or `-log(rand(·)))`. Choose a random matrix $\mathbf{G}$ and generate the corresponding $\mathbf{u}$ values as $\mathbf{u} = \mathbf{G} \cdot \mathbf{v}$. Use 2000 randomly chosen $\mathbf{v}$'s and their corresponding $\mathbf{u}'s$. Then, use independent components analysis, as in equation 10.40, to learn generative sources from the inputs $\mathbf{u}$. How well do the values of the extracted sources match those of the original sources?

9. Compare the actual $\mathbf{G}$ you used to generate the data in problem 8 with the $\mathbf{G}$ that is recovered by independent components analysis. Plot the six two-dimensional projections of the input data (u_1 versus u_2; u_1 versus u_3; *etc*) together with the projections of the mixing axes coming from $\mathbf{G}$ (it is good to use more data points for this, say 10000). The mixing axes are lines parallel to vectors with components G_{1i}, G_{2i}, G_{3i}, and G_{4i}, for $i = 1, 2, 3, 4$. What relationship exists between these mixing axes and the envelope of the data points, and why? Plot $\mathbf{u}$ generated in the same way when the components of $\mathbf{v}$ are chosen independently from identical Gaussian distributions, together with the mixing axes coming from $\mathbf{G}$. What differences do you see?

10. Implement wake-sleep learning for the Helmholtz machine with binary units when the input data is derived from a square "retina" of size `ndim`×`ndim`. The `ndim` columns of the input array are independently turned "on" with probability `pbar`. Each unit in a column that is "on" takes the value 1 with probability 1−`pout` and 0 with probability `pout`, and each unit in a column that is "off" takes the value 1 with probability `pout` and 0 with probability 1−`pout`. MATLAB® program `c10p10.m` is an example. In what way does the activity of the $\mathbf{v}$ units in the model capture the way that each input $\mathbf{u}$ was actually generated? What happens if there are not enough hidden units to represent each column separately?

11. Implement wake-sleep learning for the binary Helmholtz machine as in problem 10, except now make a correlational structure between the columns – so that for half the input patterns, only columns 1...`ndim/2` are eligible to be turned on (with probability `pbar`), and for the other half, only the other columns `ndim/2` + 1...`ndim` are eligible. Program `c10p11.m` shows one way to generate such inputs. Train a Helmholtz machine with two representational layers ($\mathbf{v}$ and z), the top layer (z) having just one unit, the middle layer ($\mathbf{v}$) with `ndim`+1 units. Does this build a generative model that captures the hierarchical way in which each input pattern is generated?