P1. Repeat in PyMC the following experiment from Murphy (2012):

Mixture of multinoullis

We can use mixture models to define density models on many kinds of data. For example, suppose our data consist of D-dimensional bit vectors. In this case, an appropriate class-conditional density is a product of Bernoullis:

$$p(x_i|z_i = k, \theta) = \prod_{j=1}^{D} Ber(x_{ij}|\mu_{jk}) = \prod_{j=1}^{D} \mu_{jk}^{x_{ij}}(1-\mu_{jk})^{1-x_{ij}}$$

$$\mu_{jk} = \text{"}the\ probability\ that\ bit\ j\ turns\ on\ in\ cluster\ k\text{"}\ (p.\,340)$$

As an example of clustering binary data, consider a binarized version of the MNIST handwriting ten digit dataset (see Figure1.5(a)), where we ignore the class labels. We can fit a mixture of Bernoullis to this, using K=10, and then visualize the resulting centroids, $\hat{\mu}_k$, as shown in Figure 11.5" (pp. 341-342).

Extras from (Murphy, 2012, p. 7):

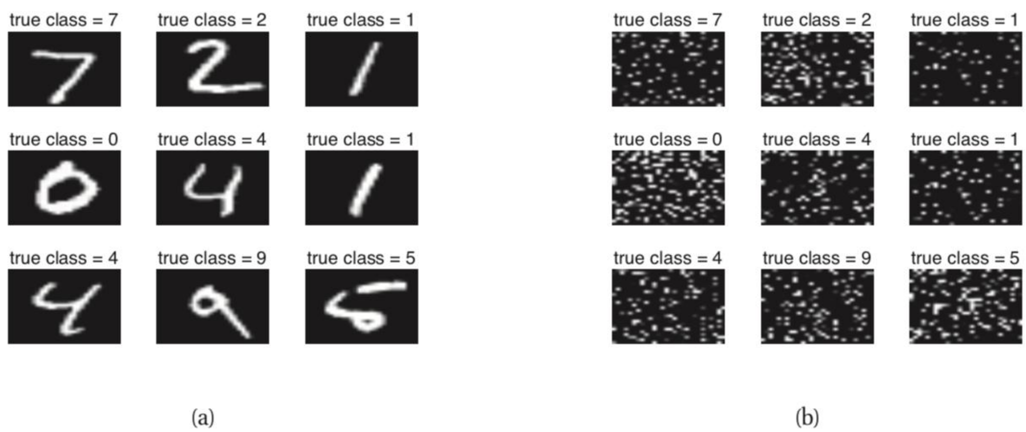

Figure 1.5 (a) First 9 test MNIST gray-scale images. (b) Same as (a), but with the features permuted randomly. Classification performance is identical on both versions of the data (assuming the training data is permuted in an identical way). Figure generated by `shuffledDigitsDemo`.

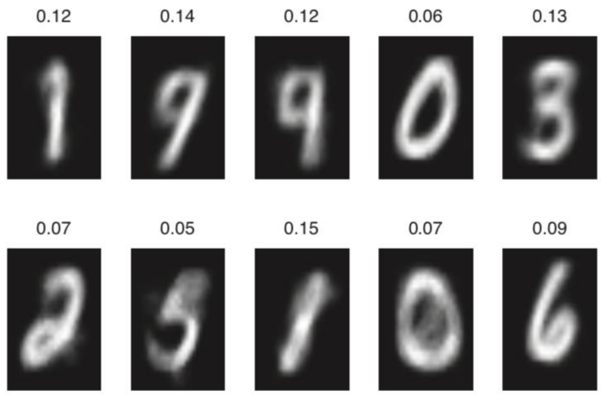

Figure 11.5 We fit a mixture of 10 Bernoullis to the binarized MNIST digit data. We show the MLE for the corresponding cluster means, $\boldsymbol{\mu}_k$. The numbers on top of each image represent the mixing weights $\hat{\pi}_k$. No labels were used when training the model. Figure generated by `mixBerMnistEM`.

CITED WORKS:

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. Cambridge: The MIT Press.