Course 67800	Spring 2021

Probabilistic Methods in Artificial Intelligence

Programming Assignment 1: Bayesian Networks

Deadline: Thursday 15/4/21, 23:59

Prof. Gal Elidan — *TA: Hagai Rappeport*

In this programming assignment we will investigate the structure of the binarized MNIST dataset of handwritten digits using Bayesian networks. The dataset contains images of handwritten digits with dimensions 28×28 (784) pixels. Consider the Bayesian network in Figure 1. The network contains two layers of variables. The variables in the bottom layer, $X_{1:784}$ denote the pixel values of the flattened image and are referred to as *manifest variables*. The variables in the top layer, Z_1 and Z_2, are referred to as *latent variables*, because the value of these variables will not be explicitly provided by the data and will have to be inferred.
For now, don't worry too much about what the latent variables represent, this will become clearer as you work with the network.

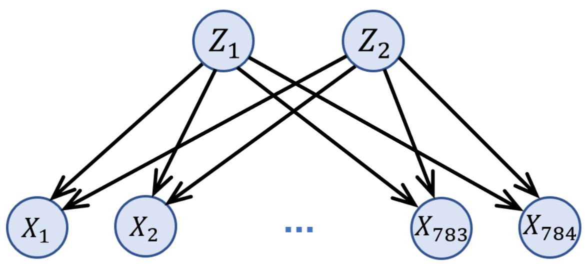

Figure 1: Bayesian network for the MNIST dataset. $X_{1:784}$ variables correspond to pixels in an image. Z_1 and Z_2 variables are latent.

The Bayesian network specifies a joint probability distribution over binary images and latent variables $p\left(Z_1, Z_2, X_{1:784}\right)$. The model is trained so that the marginal probability of the manifest variables, $p\left(X_{1:784}\right) = \sum_{z_1,z_2} p\left(z_1, z_2, X_{1:784}\right)$ is high on images that look like digits, and low for other images.

For this programming assignment, we provide a pretrained model `trained_mnist_model`. The starter code `pa1.py` loads this model and provides functions to directly access the conditional probability tables. Further, we simplify the problem by discretizing the latent and manifest variables such that $\mathrm{Val}\left(Z_1\right) = \mathrm{Val}\left(Z_2\right) = \{-3, -2.75, \ldots, 2.75, 3\}$ and $\mathrm{Val}\left(X_i\right) = \{0, 1\}$, i.e., the image is binary.

Warmup Questions

1. [5 points] How many degrees of freedom does the joint have? i.e. how many parameters would you need to specify an arbitrary probability distribution over all possible 28×28 binary images?

2. [5 points] How many degrees of freedom does the BN in fig. 1 have?

Assignment

For questions 1-4 below, refer to `pa1.py`. The starter code contains some helper functions for solving these questions. It is not compulsory to use them and you are allowed to use your own implementations. Also, feel free to introduce your own additional helper functions when useful.

Tips:

- Try to avoid long loops (e.g. over data samples, over the dimension of X). Use numpy vector operations instead. The run time is typically a few seconds, except q6 which might run for a few minutes.

- You can use basic numpy functions such as np.random.choice or np.random.uniform.

1. [25 points] Produce 10 samples from the joint probability distribution $(z_1, z_2, x_{1:784}) \sim p(Z_1, Z_2, X_{1:784})$, and plot the corresponding binary images (sampled values of $X_{1:784}$).
   Hint: they should look more or less like (binarized) handwritten digits.

2. [15 points] For each possible value of

$$(\overline{z}_1, \overline{z}_2) \in \{-3, -2.75, \ldots, 2.75, 3\} \times \{-3, -2.75, \ldots, 2.75, 3\}$$

   compute the conditional expectation $\mathbb{E}[X_{1:784} | Z_1, Z_2 = \overline{z}_1, \overline{z}_2]$. This is the expected image corresponding to each possible value of the latent variables Z_1, Z_2. Plot the images on on a 2D grid where the grid axes correspond to Z_1 and Z_2 respectively. What is the intuitive role of the Z_1, Z_2 variables in this model?

3. [25 points] In `q3.mat`, you are given a *validation* and a *test* dataset. In the *test* dataset, some images are "real" handwritten digits, and some are anomalous (corrupted images). We would like to use our Bayesian network to distinguish real images from the anomalous ones. Intuitively, our Bayesian network should assign low probability to corrupted images and high probability to the real ones, and we can use this for classification. To do this, we first compute the average and the standard deviation of the marginal log-likelihood,

$$\log p(x_{1:784}) = \log \sum_{z_1} \sum_{z_2} p(z_1, z_2, x_{1:784})$$

   on the validation dataset (average and std of $\log p(x_{1:784})$ over the validation set samples). Consider a simple prediction rule where images with marginal log-likelihood, $\log p(x_{1:784})$, below **three** standard deviations of the average marginal log-likelihood are classified as corrupted. Classify all images in the **test set** as corrupted or real using this rule. Then plot a histogram of the marginal log-likelihood for the images classified as "real". Plot a separate histogram of the marginal log-likelihood for the images classified as "corrupted".

   Hint: If you run into any flow issues, search for the "log-sum-exp trick" online for help.

4. [25 points] In `q4.mat`, you are given a labeled dataset of images of handwritten digits (the label corresponds to the digit identity). For each image I^k, compute the conditional probabilities $p\left((Z_1, Z_2) \mid X_{1:784} = I^k\right)$. Use these probabilities to compute the conditional expectation

$$\mathbb{E}\left[(Z_1, Z_2) \mid X_{1:784} = I^k\right]$$

   Plot all the conditional expectations in a single plot (a scatter plot – one point per image), color coding each point as per their label. What is the relationship with the figure you produced for Q 2?