Homework 5

Deadline: Wednesday, Mar. 20, at 11:59pm.

Submission: You need to submit two files:

1. Your solutions to Questions 1, 2, and 3 as a PDF file, `hw5_writeup.pdf`, through MarkUs[1].
2. Your completed Python code for Question 1, as `q1.py`.

Neatness Point: One of the 10 points will be given for neatness. You will receive this point as long as we don't have a hard time reading your solutions or understanding the structure of your code.

Late Submission: 10% of the marks will be deducted for each day late, up to a maximum of 3 days. After that, no submissions will be accepted.

Collaboration. Weekly homeworks are individual work. See the Course Information handout[2] for detailed policies.

1. **[3pts] Gaussian Discriminant Analysis.** For this question you will build classifiers to label images of handwritten digits. Each image is 8 by 8 pixels and is represented as a vector of dimension 64 by listing all the pixel values in raster scan order. The images are grayscale and the pixel values are between 0 and 1. The labels y are $0, 1, 2, \cdots, 9$ corresponding to which character was written in the image. There are 700 training cases and 400 test cases for each digit; they can be found in `a2digits.zip`.

   Starter code is provided to help you load the data (`data.py`). A skeleton (`q1.py`) is also provided for each question that you should use to structure your code.

   Using maximum likelihood, fit a set of 10 class-conditional Gaussians with a separate, full covariance matrix for each class. Remember that the conditional multivariate Gaussian probability density is given by,

$$p(\mathbf{x} \mid y = k, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-d/2} |\boldsymbol{\Sigma}_k|^{-1/2} \exp\left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right\} \tag{1}$$

   You should take $p(y = k) = \frac{1}{10}$. You will compute parameters μ_{kj} and $\boldsymbol{\Sigma}_k$ for $k \in \{0, \ldots, 9\}, j \in \{1, \ldots, 64\}$. You should implement the covariance computation yourself (i.e. without the aid of 'np.cov'). To ensure numerical stability you will have to add a small multiple of the identity to each covariance matrix. For this assignment you should add $0.01\mathbf{I}$ to each $\boldsymbol{\Sigma}_k$ after computing its maximum likelihood estimate.

   (a) **[1pt]** Using the parameters you fit on the training set and Bayes rule, compute the average conditional log-likelihood, i.e. $\frac{1}{N} \sum_{i=1}^{N} \log(p(y^{(i)} \mid \mathbf{x}^{(i)}, \theta))$ on both the train and test set and report it.

   (b) **[1pt]** Select the most likely posterior class for each training and test data point as your prediction, and report your accuracy on the train and test set.

[1] `https://markus.teach.cs.toronto.edu/csc411-2019-01`

[2] `http://www.cs.toronto.edu/~mren/teach/csc411_19s/syllabus.pdf`

(c) **[1pt]** Compute the leading eigenvectors (largest eigenvalue) for each class covariance matrix (can use `np.linalg.eig`) and plot them side by side as 8 by 8 images.

Report your answers to the above questions, and submit your completed Python code for `q1.py`.

2. **[2pts] Categorial Distribution.** Let's consider fitting the categorical distribution, which is a discrete distribution over K outcomes, which we'll number 1 through K. The probability of each category is explicitly represented with parameter θ_k. For it to be a valid probability distribution, we clearly need $\theta_k \geq 0$ and $\sum_k \theta_k = 1$. We'll represent each observation $\mathbf{x}$ as a 1-of-K encoding, i.e, a vector where one of the entries is 1 and the rest are 0. Under this model, the probability of an observation can be written in the following form:

$$p(\mathbf{x}; \boldsymbol{\theta}) = \prod_{k=1}^{K} \theta_k^{x_k}.$$

Suppose we observe a dataset $\mathcal{D} = \{\mathbf{x}^{(n)}\}_{n=1}^{N}$. Denote the count for outcome k as N_k, so $N_k = \sum_{n=1}^{N} x_k^{(n)}$. In the previous assignment, you showed that the maximum likelihood estimate for the counts was:

$$\hat{\theta}_k = \frac{N_k}{N}.$$

Now let's derive the Bayesian parameter estimate.

(a) **[1pts]** For the prior, we'll use the Dirichlet distribution, which is defined over the set of probability vectors (i.e. vectors that are nonnegative and whose entries sum to 1). Its PDF is as follows:

$$p(\boldsymbol{\theta}) \propto \theta_1^{a_1 - 1} \cdots \theta_K^{a_k - 1}.$$

A useful fact is that if $\boldsymbol{\theta} \sim \text{Dirichlet}(a_1, \ldots, a_K)$, then

$$\mathbb{E}[\theta_k] = \frac{a_k}{\sum_{k'} a_{k'}}.$$

Determine the posterior distribution $p(\boldsymbol{\theta} \,|\, \mathcal{D})$. From that, determine the posterior predictive probability that the next outcome will be k.

(b) **[1pt]** Still assuming the Dirichlet prior distribution, determine the MAP estimate of the parameter vector $\boldsymbol{\theta}$. For this question, you may assume each $a_k > 1$.

3. **[4pts] Factor Analysis.**

In lecture, we covered the EM algorithm applied to mixture of Gaussians models. In this question, we'll look at another interesting example of EM, namely factor analysis. This is a model very similar in spirit to PCA: we have data in a high-dimensional space, and we'd like to summarize it with a lower-dimensional representation. Unlike PCA, we formulate the problem in terms of a probabilistic model. We assume the latent code vector $\mathbf{z}$ is drawn from a standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, and that the observations are drawn from a diagonal covariance Gaussian whose mean is a linear function of $\mathbf{z}$. We'll consider the slightly simplified case of scalar-valued z. The probabilistic model is given by:

$$z \sim \mathcal{N}(0, 1)$$
$$\mathbf{x} \,|\, z \sim \mathcal{N}(z\mathbf{u}, \boldsymbol{\Sigma}),$$

where $\mathbf{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_D^2)$. Note that the observation model can be written in terms of coordinates:

$$x_d \,|\, z \sim \mathcal{N}(z u_d, \sigma_j).$$

We have a set of observations $\{\mathbf{x}^{(n)}\}_{n=1}^N$, and z is a latent variable, analogous to the mixture component in a mixture-of-Gaussians model. The parameters of the model are $\boldsymbol{\theta} = \{\mathbf{u}, \mathbf{\Sigma}\}$.

In this question, we'll derive both the E-step and the M-step for the EM algorithm. If you don't feel like you understand the EM algorithm yet, don't worry; we'll walk you through it, and the question will be mostly mechanical.

(a) **E-step (2pts).** In this step, our job is determine the posterior distribution $q(z) = p(z \,|\, \mathbf{x}; \boldsymbol{\theta})$. After finding this distribution, you should compute formulas for the (univariate) statistics which we'll use in the M-step:

$$m = \mathbb{E}[z \,|\, \mathbf{x}; \boldsymbol{\theta}] =$$
$$s = \mathbb{E}[z^2 \,|\, \mathbf{x}; \boldsymbol{\theta}] =$$

Tips:

- Compare the model here with the linear Gaussian model of the Appendix. Note that z here is a scalar, while the Appendix gives the more general formulation where $\mathbf{x}$ and $\mathbf{z}$ are both vectors.
- To help you check your work: $p(z \,|\, \mathbf{x}; \boldsymbol{\theta})$ is a univariate Gaussian distribution whose mean is a linear function of $\mathbf{x}$, and whose variance does not depend on $\mathbf{x}$.
- Once you have figured out the mean and variance, that will give you the conditional expectations.

(b) **M-step (2pts).** In this step, we need to re-estimate the parameters of the model. Suppose the current parameters (used in the E-step) are $\boldsymbol{\theta}_{\text{old}} = \{\mathbf{u}_{\text{old}}, \mathbf{\Sigma}_{\text{old}}\}$. For this part, your job is to derive a formula for $\mathbf{u}_{\text{new}}$ that maximizes the expected log-likelihood, i.e.,

$$\mathbf{u}_{\text{new}} \leftarrow \arg\max_{\mathbf{u}} \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}_{q_n(z^{(n)})}[\log p(z^{(n)}, \mathbf{x}^{(n)}; \boldsymbol{\theta})].$$

Recall that $q_n(z^{(n)}) = p(z^{(n)} | \mathbf{x}^{(n)}; \boldsymbol{\theta}_{\text{old}})$ is the distribution computed in part (a). Your answer should be given in terms of the $m^{(n)} = \mathbb{E}[z^{(n)} \,|\, \mathbf{x}^{(n)}; \boldsymbol{\theta}_{\text{old}}]$ and $s^{(n)} = \mathbb{E}[(z^{(n)})^2 \,|\, \mathbf{x}^{(n)}; \boldsymbol{\theta}_{\text{old}}]$ from the previous part. (I.e., you don't need to expand out the formulas for $m^{(n)}$ and $s^{(n)}$ in this step, because if you were implementing this algorithm, you'd use the values $m^{(n)}$ and $s^{(n)}$ that you previously computed.)

Tips:

- Expand $\log p(z^{(n)}, \mathbf{x}^{(n)}; \boldsymbol{\theta})$ to $\log p(z^{(n)}; \boldsymbol{\theta}) + \log p(\mathbf{x}^{(n)} \mid z^{(n)}; \boldsymbol{\theta})$ (log is the natural logarithm).
- Expand out the PDF of the Gaussian distribution.
- Apply linearity of expectation. You should wind up with terms proportional to $\mathbb{E}_{q_n(z^{(n)})}[z^{(n)}]$ and $\mathbb{E}_{q_n(z^{(n)})}[(z^{(n)})^2]$. Replace these expectations with $m^{(n)}$ and $s^{(n)}$. You should get an equation that does not mention $z^{(n)}$.
- In order to find the maximum likelihood parameter $\mathbf{u}_{\text{new}}$, you need to take the derivative with respect to u_d, set it to zero, and solve for $\mathbf{u}_{\text{new}}$.

(c) **M-step, cont'd (optional)** Find the M-step update for the observation variances $\{\sigma_d\}_{d=1}^D$. This can be done in a similar way to part (b).

Appendix: Some Properties of Conditional Gaussians

Consider a multivariate Gaussian random variable $\mathbf{z}$ with the mean $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Lambda}^{-1}$ ($\boldsymbol{\Lambda}$ is the inverse of the covariance matrix and is called the precision matrix). We denote this by

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z} \,|\, \boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}).$$

Now consider another Gaussian random variable $\mathbf{x}$, whose mean is an affine function of $\mathbf{z}$ (in the form to be clear soon), and its covariance $\mathbf{L}^{-1}$ is independent of $\mathbf{z}$. The conditional distribution of $\mathbf{x}$ given $\mathbf{z}$ is

$$p(\mathbf{x} \,|\, \mathbf{z}) = \mathcal{N}(\mathbf{x} \,|\, \mathbf{A}\mathbf{z} + \mathbf{b}, \mathbf{L}^{-1}).$$

Here the matrix $\mathbf{A}$ and the vector $\mathbf{b}$ are of appropriate dimensions.

In some problems, we are interested in knowing the distribution of $\mathbf{z}$ given $\mathbf{x}$, or the marginal distribution of $\mathbf{x}$. One can apply Bayes' rule to find the conditional distribution $p(\mathbf{z} \,|\, \mathbf{x})$. After some calculations, we can obtain the following useful formulae:

$$p(\mathbf{x}) = \mathcal{N}\left(\mathbf{x} \,|\, \mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A}\boldsymbol{\Lambda}^{-1}\mathbf{A}^\top\right)$$
$$p(\mathbf{z} \,|\, \mathbf{x}) = \mathcal{N}\left(\mathbf{z} \,|\, \mathbf{C}(\mathbf{A}^\top\mathbf{L}(\mathbf{x} - \mathbf{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu}), \mathbf{C}\right)$$

with

$$\mathbf{C} = (\boldsymbol{\Lambda} + \mathbf{A}^\top\mathbf{L}\mathbf{A})^{-1}.$$