HOMEWORK 3

CSC2515 Fall 2021

- **Deadline:** Friday, December 10, 2021 at **16:59**.
- **Submission:** You need to submit two files through MarkUs. One is a PDF file including all your answers and plots. The other is a source file (Python script or Jupyter Notebook) that reproduces your answers. You can produce the file however you like (e.g. LaTeX, Microsoft Word, etc) as long as it is readable. Points will be deducted if we have a hard time reading your solutions or understanding the structure of your code. If the code does not run, you may lose most/all of your points for that question.
- **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:** You can discuss the assignment with up to two other students (group of three). You can work on the code together. But each of you need to **write your homework report individually**. You must mention the name of your collaborators clearly in the report and the source code.

1. Class-Conditional Gaussians – 30 pts. In this question, you will derive the maximum likelihood estimates for class-conditional Gaussians with independent features (diagonal covariance matrices), i.e., Gaussian Naïve Bayes, with shared variances.

Start with the following generative model for a discrete class label $y \in \{1, 2, ..., k\}$ and a real-valued vector of d features $\mathbf{x} = (x_1, x_2, ..., x_d)$:

$$p(y = k) = \alpha_k, \tag{1.1}$$

$$p(\mathbf{x}|y = k, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \left(\prod_{i=1}^{D} 2\pi\sigma_i^2\right)^{-1/2} \exp\left(-\sum_{i=1}^{D} \frac{(x_i - \mu_{ki})^2}{2\sigma_i^2}\right), \tag{1.2}$$

where α_k is the prior on class k, σ_i^2 are the shared variances for each feature (common for all classes), and μ_{ki} is the mean of the feature i conditioned on class k. We use $\boldsymbol{\alpha}$ to denote the vector with elements α_k. Similarly, $\boldsymbol{\sigma}$ denotes the vector of variances. The matrix of class means is written $\boldsymbol{\mu}$ where the kth row of $\boldsymbol{\mu}$ is the mean for class k.

1. **[4pt]** Use Bayes' rule to derive an expression for $p(y = k|x, \boldsymbol{\mu}, \boldsymbol{\sigma})$. [**Hint**: Use the law of total probability to derive an expression for $p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\sigma})$.]

2. **[8pt]** Write down the expression for the negative likelihood function (NLL)

$$\ell(\boldsymbol{\theta}; D) = -\log p\Big(y^{(1)}, \mathbf{x}^{(1)}, y^{(2)}, \mathbf{x}^{(2)}, \cdots, y^{(N)}, \mathbf{x}^{(N)} \mid \boldsymbol{\theta}\Big) \tag{1.3}$$

of a particular dataset $D = \{(y^{(1)}, \mathbf{x}^{(1)}), (y^{(2)}, \mathbf{x}^{(2)}), \cdots, (y^{(N)}, \mathbf{x}^{(N)})\}$ with parameters $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\sigma}\}$ (assume that the data points are i.i.d.)

3. **[10pt]** Take partial derivatives of the likelihood with respect to each of the parameters μ_{ki} and with respect to the shared variances σ_i^2.
4. **[8pt]** Find the maximum likelihood estimates for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$.

5*. Extra work for interested students: If you are familiar with Lagrange multipliers, show that the MLE for α_k is indeed given by (1.4):

$$\alpha_k = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}[y^{(i)} = k] \tag{1.4}$$

2. Handwritten Digit Classification with Generative Models – 70 pts. For this question you will build generative 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 a4digits.zip.

Starter code written in Python is provided to help you load the data. A skeleton is also provided for each question that you should use to format your solution.

Please note that if you are asked to report/compute quantities these should be clearly displayed in your written report. It is not sufficient to simply print these as an output of your code. The same applies to plots and figures.

0. **[2pt]** Load the data and plot the means for each of the digit classes in the training data (include these in your report). Given that each image is a vector of size 64, the mean will be a vector of size 64 which needs to be reshaped as an 8×8 2D array to be rendered as an image. Plot all 10 means side by side using the same scale.

2.1. *Conditional Gaussian Classifier Training – 30 pts.* 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

$$p(\mathbf{x}|y = k, \boldsymbol{\mu}, \Sigma_k) = (2\pi)^{-d/2}|\Sigma_k|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\right). \tag{2.1}$$

You should take $p(y = k) = \frac{1}{10}$. You will compute parameters μ_{kj} and Σ_k for $k \in \{0, \ldots, 9\}$ and $j \in \{1, \ldots, 64\}$. You should implement the covariance computation yourself, i.e., without the aid of 'np.cov'. [**Hint:** To ensure numerical stability you may have to add a small positive value to the diagonal of each covariance matrix. For this assignment you can add $0.01\mathbf{I}$ to each matrix.]

1. **[18pt]** Plot an 8 by 8 image of the log of the diagonal elements of each covariance matrix Σ_k. Plot all ten classes side by side using the same grayscale.
2. **[6pt]** Using the parameters you fit on the training set and the Bayes' rule, compute the average conditional log-likelihood, i.e. $\frac{1}{N}\sum_{i=1}^{N} \log p(y^{(i)}|\mathbf{x}^{(i)}, \theta)$ on both the train and test set and report it. (Here θ denotes all the estimated parameters.)
3. **[6pt]** 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.

4*. Extra work for interested students: Compute the leading eigenvectors (largest eigenvalue) for each class covariance matrix (you can use np.linalg.eig) and plot them side by side as 8 by 8 images.

2.2. *Naive Bayes Classifier Training – 35 pts.*

1. **[1pt]** Convert the real-valued features $\mathbf{x}$ into binary features $\mathbf{b}$ using 0.5 as a threshold: $b_j = 1$ if $x_j > 0.5$ otherwise $b_j = 0$.
2. **[15pt]** Using these new binary features $\mathbf{b}$ and the class labels, train a Bernoulli Naïve Bayes classifier using MAP estimation with prior $\mathsf{Beta}(\alpha, \beta)$ with $\alpha = \beta = 2$. In particular, fit the model below on the training set.

$$p(y = k) = \frac{1}{10} \tag{2.2}$$

$$p(b_j = 1|y = k) = \eta_{kj} \tag{2.3}$$

$$p(\mathbf{b}|y = k, \eta) = \Pi_{j=1}^{d}(\eta_{kj})^{b_j}(1 - \eta_{kj})^{(1-b_j)} \tag{2.4}$$

$$p(\eta_{kj}) = \mathsf{Beta}(2, 2) \tag{2.5}$$

You should compute parameters η_{kj} for $k \in \{0, \ldots, 9\}$ and $j \in \{1, \ldots, 64\}$.

Prior as Pseudo-Counts:. Instead of explicitly considering the Beta distribution prior in the Bernoulli likelihood model, you can add two training cases to your data set for each class, one of which has every pixels OFF and the other has every pixels ON. Make sure you understand why this is equivalent to using a prior. You may use either schemes in your own code.

3. **[3pt]** Plot each of your $\boldsymbol{\eta}_k$ vectors as an 8 by 8 grayscale image. These should be presented side by side and with the same scale.
4. **[6pt]** Given your parameters, sample one new data point using your generative model for each of the 10 digit classes. Plot these new data points as 8 by 8 grayscale images side by side.
5. **[6pt]** Using the parameters you fit on the training set and Bayes' rule, compute the average conditional log-likelihood, i.e. $\frac{1}{N}\sum_{i=1}^{N} \log p(y^{(i)}|\mathbf{x}^{(i)}, \theta)$ on both the train and test set and report it.
6. **[4pt]** Select the most likely posterior class for each training and test data point, and report your accuracy on the train and test set.

2.3. *Model Comparison – 3 pts.* Briefly (in a few sentences) summarize the performance of each model. Which performed best? Which performed worst? Did this match your expectations?