CSC2515：Assignment #2
Due: 10pm, 12th November

YIFAN ZHANG

Student Number: 1004245952

November 12, 2017

Problem 1: Class-Conditional Gaussians

In this question, you will derive the maximum likelihood estimates for class-conditional Gaussians with independent features(diagonal covariance matrices), i.e. Gaussian Na ïve 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 $\mathbf{x} = (x_1, x_2, ..., x_d)$:

$$p(y = k) = \alpha_k \qquad (1)$$

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

Where α_k is the prior on class k, σ_i^2 are the shared variances for each feature (in all classes), and μ_{ki} is the mean of the feature i conditioned on class k . We write to represent the vector with elements α_k and similarity is the vector of variances $\boldsymbol{\sigma}$. The matrix of class mean is written $\boldsymbol{\mu}$ where the k th row of $\boldsymbol{\mu}$ is the mean for class k .

1. Use Bayes 'rules to derive an expression for $p(y = k \mid \mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\sigma})$

$$p(y = k \mid \mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{p(\mathbf{x} \mid y = k, \boldsymbol{\mu}, \boldsymbol{\sigma}) p(y = k)}{p(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\sigma})}$$

By using the law of total probability:

$$p(y = k \mid \mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{p(\mathbf{x} \mid y = k, \boldsymbol{\mu}, \boldsymbol{\sigma}) p(y = k)}{\sum_{t=1}^{K} p(\mathbf{x} \mid y = t, \boldsymbol{\mu}, \boldsymbol{\sigma}) p(y = t)}$$

$$= \frac{(\prod_{i=1}^{d} 2\pi\sigma_i^2)^{-1/2} \exp\{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i - \mu_{ki})^2\}\alpha_k}{\sum_{t=1}^{K}(\prod_{i=1}^{d} 2\pi\sigma_i^2)^{-1/2} \exp\{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i - \mu_{ti})^2\}\alpha_t}$$

2. Write down an expression for the negative likelihood function(NLL)

$$l(\theta; D) = -\log p(y^{(1)}, \mathbf{x}^{(1)}, y^{(2)}, \mathbf{x}^{(2)}, ..., y^{(N)}, \mathbf{x}^{(N)} \mid \boldsymbol{\theta}) \qquad (3)$$

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

Since we assume that the data are independent and identically distributed random variables, we could rewrite $l(\theta; D)$ as following:

$$\ell(\theta; D) = -\log p(y^{(1)}, \mathbf{x}^{(1)}, y^{(2)}, \mathbf{x}^{(2)}, ..., y^{(N)}, \mathbf{x}^{(N)} \mid \boldsymbol{\theta})$$
$$= -\log \prod_{j=1}^{N} p(y^{(j)}, \mathbf{x}^{(j)} \mid \boldsymbol{\theta})$$
$$= -\sum_{j=1}^{N} \log(p(y^{(j)}, \mathbf{x}^{(j)} \mid \boldsymbol{\theta}))$$
$$= -\sum_{i=1}^{N} [\log p(y^{(i)}) + \log p(\mathbf{x}^{(i)} \mid y^{(i)}, \boldsymbol{\theta}^{(i)})]$$

$$p(\mathbf{x}^{(j)} \mid y^{(j)}, \boldsymbol{\theta}) = (\prod_{i=1}^{d} 2\pi\sigma_i^2)^{-1/2} \exp\{-\sum_{i=1}^{d} \frac{1}{2\sigma_i^2} (x_i^{(j)} - \mu_{ki})^2\}$$

$$\log p(\mathbf{x}^{(j)} \mid y^{(j)}, \boldsymbol{\theta}) = -\frac{1}{2} \sum_{i=1}^{d} \log(2\pi\sigma_i^2) - \sum_{i=1}^{d} \frac{1}{2\sigma_i^2} (x_i^{(j)} - \mu_{ki})^2$$

$$\ell(\theta; D) = \sum_{j=1}^{N} \sum_{i=1}^{d} \frac{1}{2} [\log(2\pi\sigma_i^2) + \frac{1}{\sigma_i^2} (x_i^{(j)} - \mu_{ki})^2] - \log \alpha_k$$

3. 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 .

$$\frac{\partial \ell(\theta; D)}{\partial \mu_{ki}} = \sum_{j=1}^{N} \mathbf{1}(y^{(i)} = k) \frac{1}{\sigma_i^2} (\mu_{ki} - x_i^{(j)})$$

$$\frac{\partial \ell(\theta; D)}{\partial \sigma_i^2} = \sum_{j=1}^{N} \mathbf{1}(y^{(i)} = k) \frac{1}{2} [\frac{1}{\sigma_i^2} - \frac{1}{\sigma_i^4} (\mu_{ki} - x_i^{(j)})^2]$$

Where $\mathbf{1}(y^{(i)} = k) = \begin{cases} 1 & if \quad y^{(i)} = k \\ 0 & otherwise \end{cases}$

4. Find the maximum likelihood estimates for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$.

To maximize $\ell(\theta; D)$, we have $\frac{\partial \ell(\theta; D)}{\partial \mu_{ki}} = \sum_{j=1}^{N} \mathbf{1}(y^{(i)} = k) \frac{1}{2\sigma_i^2} (\mu_{ki} - x_i^{(j)}) = 0$

$$\mu_{ki} = \frac{\sum_{j=1}^{N} \mathbf{1}(y^{(j)} = k) x_i^{(j)}}{\sum_{j=1}^{N} \mathbf{1}(y^{(j)} = k)}$$

$$\boldsymbol{\mu}_k = \frac{\sum_{j=1}^{N} \mathbf{1}(y^{(j)} = k)\mathbf{x}^{(j)}}{\sum_{j=1}^{N} \mathbf{1}(y^{(j)} = k)}$$

In the same way, we have $\frac{\partial \ell(\theta; D)}{\partial \sigma_i^2} = \sum_{j=1}^{N} \mathbf{1}(y^{(i)} = k)\frac{1}{2}[\frac{1}{\sigma_i^2} - \frac{1}{\sigma_i^4}(\mu_{ki} - x_i^{(j)})^2]=0$

$$\sigma_i{}^2 = \frac{\sum_{j=1}^{N} \mathbf{1}(y^{(j)} = k)(x_i^{(j)} - \mu_{ki})^2}{\sum_{j=1}^{N} \mathbf{1}(y^{(j)} = k)}$$

$$\boldsymbol{\sigma}^2 = \frac{\sum_{j=1}^{N} \mathbf{1}(y^{(j)} = k)(\mathbf{x}^{(j)} - \boldsymbol{\mu}_k)^2}{\sum_{j=1}^{N} \mathbf{1}(y^{(j)} = k)}$$

$$\boldsymbol{\sigma} = \sqrt{\frac{\sum_{j=1}^{N} \mathbf{1}(y^{(j)} = k)(\mathbf{x}^{(j)} - \boldsymbol{\mu}_k)^2}{\sum_{j=1}^{N} \mathbf{1}(y^{(j)} = k)}}$$

Problem 2: Handwritten Digit Classification

2.0 Load Data

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.

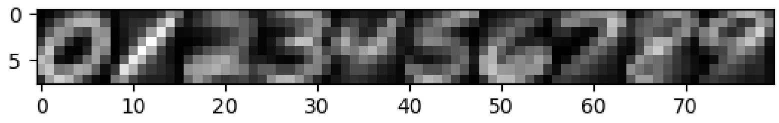

Image 1 the means for each of the digit classes in the training data

2.1 K-NN Classifier

1. Build a simple K nearest neighbor classifier using Euclidean distance on the raw pixel data.
   (a) For K = 1 report the train and test classification accuracy.

   When K = 1, the train accuracy is 1.0, the test accuracy is 0.96875.

   (b) For K = 15 report the train and test classification accuracy.

   When K = 15, the train accuracy is 0.9594285714285714, the test accuracy is 0.9585.

2. For K > 1 K-NN might encounter ties that need to be broken in order to make a decision. Choose any (reasonable) method you prefer and explain it briefly in your report.

   When ties occurs, we could decrease the value of K until the tie is broken. If it doesn't work, use the class given a 1NN classifier. When we choose K=1, there will be no ties when choose the class.

3. Use 10 fold cross validation to find the optimal K in the 1-15 range. You may use the KFold implementation in sklearn or your existing code from Assignment 1. Report this value of K along with the train classification accuracy, the average accuracy across folds and the test accuracy.

As shown in Table1, the optimal K is 1, and its test accuracy is 0.9689.

KNN	TRAIN ACCURACY	VALIDATION ACCURACY
1	1.0	0.964428571429
2	0.98173015873	0.957571428571
3	0.982507936508	0.963428571429
4	0.976952380952	0.961
5	0.976634920635	0.960857142857
6	0.972952380952	0.959
7	0.971936507937	0.957857142857
8	0.968825396825	0.957428571429
9	0.967206349206	0.955571428571
10	0.965333333333	0.952857142857
11	0.963666666667	0.952285714286
12	0.961666666667	0.951428571429
13	0.960365079365	0.950428571429
14	0.958111111111	0.95
15	0.957428571428	0.948764214768

Table 2 10 fold cross validation

2.2 Conditional Gaussian Classifier Training

1. Plot an 8 by 8 image of the log of the diagonal elements of each covariance matrix $\sum_k$.

   Plot all ten classes side by side using the same grayscale.

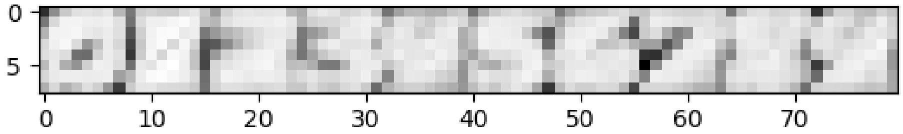

Image 2 the log of the diagonal elements of each covariance matrix $\sum_k$

2. 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.

	Train set	*Test set*
Class 0	-0.55765323	-1.01660556
Class 1	-0.17828708	-2.18786422
Class 2	-0.16591792	-0.13531796
Class 3	-0.23751501	-0.92961064
Class 4	-0.2005425	-0.34815886
Class 5	-0.76673969	-1.55992474
Class 6	-0.16503264	-1.5343051
Class 7	-1.56109209	-5.61468263
Class 8	-0.17684204	-0.54047012
Class 9	-0.50541123	-1.87390801

Table 2 average conditional log-likelihood on train set and test set

3. 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.

The accuracy on train set is 0.98025.

The accuracy on test set is 0.95925.

2.3 Naïve Bayes Classifier Training

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

$$p(y=k)=\frac{1}{10} \tag{6}$$

$$p(b_j=1 \mid y=k)=\eta_{ik} \tag{7}$$

$$p(\mathbf{b} \mid y=k,\eta)=\prod_{j=1}^{d}(\eta_{kj})^{b_j}(1-\eta_{kj})^{(1-b_j)} \tag{8}$$

$$P(\eta_{kj})=Beta(2,2) \tag{9}$$

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

Regularization: Instead of the prior, you could add two training cases to your data set for each class, one which has every pixel off and one which has every pixel on. Make sure you understand why this is equivalent to using a prior. You may use either scheme in your own code.

3. Plot each of your vectors $\boldsymbol{\eta}_k$ as an 8 by 8 grayscale image. These should be presented side by side and with the same scale.

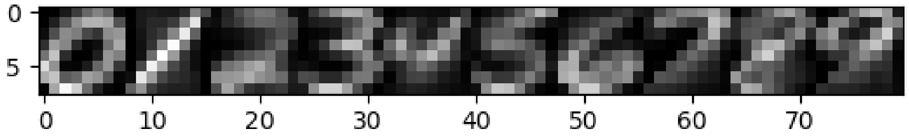

Image 3 vectors $\boldsymbol{\eta}_k$

4. Given your parameters, sample one new data point for each of the 10 digit classes. Plot these new data points as 8 by 8 grayscale images side by side.

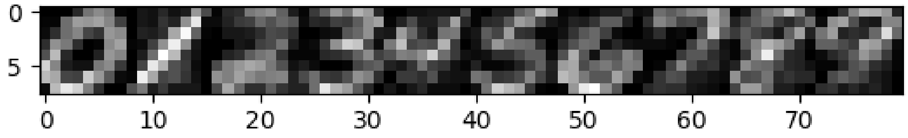

Image 4 new data point for each of the 10 digit classes

5. 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.

	Train set	*Test set*
Class 0	-0.59446351	-0.80802978
Class 1	-1.64538374	-1.36832065
Class 2	-0.95763971	-1.04132447

Class 3	-0.81003522	-0.99495436
Class 4	-0.71168767	-0.7376606
Class 5	-0.82672654	-0.90549662
Class 6	-0.63869742	-0.76775494
Class 7	-0.88781893	-0.94937512
Class 8	-1.16369827	-1.13435778
Class 9	-1.23605574	-1.19641002

Table 3 average conditional log-likelihood on train set and test set

6. Select the most likely posterior class for each training and test data point, and report your accuracy on the train and test set.

   The accuracy on train set is 0.78375.

   The accuracy on test set is 0.76525

2.4 Model Comparison

Both K-NN Classifier and Conditional Gaussian Classifier performed well, the accuracy of training on train set and test set could reach 95% (even higher). However, the speed of K-NN is not quick as Conditional Gaussian and the accuracy will decrease when we choose a larger number of neighbors.

Naïve Bayes Classifier performed worst. The accuracy of it only reach about 77% on test set and 78% on train set. This match my expectation. Because we assume conditional independence of each dimension of x. But in fact, there is correlations between dimensions and it will have a highly negative influence on classification.