In the Name of God

Statistical Pattern Recognition

Homework II

Multi-class and Binary Classification using Logistic Regression and Bayesian Classifiers

Assignment Date: **14 Aban** Submission Deadline: **27 Aban**

Contents

1 Objectives and Precautions 2

2 Multi-class Classification using Logistic Regression 3

2.1 Understanding the data: data.txt 3

2.2 Creating our model: OvO, OvA, Softmax Regression 3

3 Binary Classification using Bayesian Classifiers 4

3.1 Understanding the data: UCI Mushroom Data 4

3.2 Creating our model: LDA And QDA 4

4 Bonus 6

4.1 The Naive Bayes Algorithm 6

1 Objectives and Precautions

In this homework you will learn:

- More on how to understand a dataset (so called EDA: Explanatory Data Analysis)
- Implement multi-class classification using the logistic regression family of algorithms
- Implement binary classification using Bayesian Classifiers

Keep in mind that you can only use these python libraries in your implementations, unless mentioned in the homework instructions:

- Pandas, Numpy, Matplotlib.pyplot, Scipy, and Seaborn (for heat maps if necessary)
- And from sklearn you can only use: train_test_split and labelEncoder (to encode categorical data into numerical)

Also note that:

- You have 2 weeks to complete this homework.
- The home work instructions and datasets will be shared with you in your **Quera class**.
- Save your plots, results, and answers in any format you want, this will be your report.
- Save your code files and your report as a zip file and upload in Quera. (naming format is "your names.zip" for example "Achraf_Hakimi_Mohamed_Salah.zip")
- Late submission strategy is: 70 percent score for one day delay and 50 percent for 2 days delay. Submissions after 2 days will not be graded.
- Use only the python programming language.
- Feel free to ask your questions in the telegram channel.
- Do not copy other works, write your own code.

Thank you. Good Luck.

2 Multi-class Classification using Logistic Regression

Discriminative Multi-class Classification focuses on learning a decision boundary that discriminates between different classes in a dataset. Three common methods for discriminative multi-class classification are one-vs-one (OvO), one-vs-all (OvA), and softmax regression, three of which you will implement in the first part of this homework.

In OvO, multiple binary classifiers are trained, each distinguishing between two classes, and the class with the most favorable outcome is chosen.

In OvA, a binary classifier is trained for each class against the rest, and the class with the highest confidence is selected.

And softmax regression, also known as multinomial logistic regression, generalizes the logistic regression to multiple classes by assigning a probability distribution over all classes for each input and choosing the class with the highest probability, making it a popular choice for multi-class classification problems.

2.1 Understanding the data: data.txt

1. Load the dataset ("data.txt“).
2. Report the number of features and classes.
3. Visualize your data in 2D planes with a combination of every two features. For example, if your dataset contains 3 features, you will have 3x3 different plots: (x:feature 1, y:feature 2), (x:feature 1, y:feature 3), (x:feature 2, y:feature 1), (x:feature 2, y:feature 3), (x:feature 3, y:feature 1), (x:feature 3, y:feature 2).
4. Use Z-score metric to detect outliers. Set threshold equal to 2.75. (You can use the Scipy library for this purpose)
5. Report the outliers as well as their counts for each class.
6. Now change the z-score threshold to 2.5 and 3. How does it impact the outliers you can detect?
7. Now look back at your plot from item 3, could you detect the outliers visually?
8. What other methods can we use to detect the outliers? Explain another method.
9. Split your data into training and test sets. Consider 75% of your data as training set and 25% as your test set. (You can use sklearn to split your data)

2.2 Creating our model: OvO, OvA, Softmax Regression

1. First implement OvO and OvA algorithms to classify your multi-class dataset. Use logistic regression as the main classifier.
2. Report train and test accuracy for both methods.
3. Plot cost function each time you use a logistic regression while implementing OvO and OvA. Also report the convergence iteration of each one.
4. Finally implement the softmax regression algorithm to classify your dataset. Also report train and test accuracy and plot the cost function.
5. Compare these three classification methods. Which one has the best performance?
6. Do you get better classification performance (better accuracy) by omitting the outliers in any of these three algorithms?

3 Binary Classification using Bayesian Classifiers

Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) are both considered to be Bayesian classifiers. They are both generative models that make assumptions about the distribution of the data, and they aim to find decision boundaries that separate different classes based on the estimated probabilities.

In the case of LDA, it assumes that the data within each class follows a multivariate Gaussian distribution with a shared covariance matrix, while QDA assumes that each class has its own covariance matrix, making it more flexible but also potentially requiring more parameters.

3.1 Understanding the data: UCI Mushroom Data

The UCI Mushroom dataset contains information about different types of mushrooms, with a focus on whether they are edible or poisonous.

It contains various features that describe the physical characteristics of mushrooms, including cap shape, cap color, gill size, odor, habitat, and many others. These features are categorical, and some are binary, while others have multiple categories.

Here we are interested in classification of mushrooms into two classes of edible or poisonous, using Bayesian classifiers. Follow these instructions to understand and prepare your dataset for the modeling section:

1. Load the mushroom dataset from this **url**, and rename the features using the names located in the "mushroom_feature_names.csv" file.
2. Check the datatype of each featue in your dataset.
3. How many unique values does each feature have? What are the unique values for features: cap-color, class, and veil-type?
4. Later in this homework you will need to implement the Mahalanobis distance for your Bayesian classifier. Can you calculate any kind of distance between your samples now? why?
5. Encode your features from chategorical to numerical using the labelEncoder API of the sklearn library. What are the values of cap-color now?
6. Finally check your dataset for any nan values. What is a nan value? How many nan values do you have per feature?

Now your dataset is perhaps ready for modeling. Note that We typically want our dataset code to be decoupled from our model training code, for better readability and modularity.

3.2 Creating our model: LDA And QDA

In order to create your classifier models, follow these instructions:

1. Separate your dataset into feature (X) and labels(y). Here X will be your feature matrix.
2. In order to evaluate you model, separate 20 percent as test set and the rest as training set.
3. Now before implementing your model, what is the Bayes rule? How is it used in Bayesian classification? What is each term of this rule called?
4. What are the model parameters of LDA and QDA classifiers? What is the role of each parameter? Between QDA and LDA, which one has potentially more parameters? why? (We were asked to answer sth like this in our final exam)
5. Now that you are familiar with the parameters and differences of LDA and QDA, implement them from scratch. (Meaning learn the model parameters from training samples)

6. Then classify the test and training sets using both LDA and QDA. (use the Mahalanobis distance to predict classes for unseen data, for more detail refer to equations 4.4, 4.11, 4.12 in Dr Ahamdi and Dr Azimifar's book)

7. One parameter of your models is the covariance matrix derived from the feature matrices (X). This covariance matrix though, turns out to be singular. Now what is a singular matrix? What causes the covariance matrix derived from a feature matrix to be singular? name three reasons.

8. One reason for the covariance matrix to be singular is zero or very low variance over one feature among the feature vectors (X). Meaning that this feature have the same value among all samples. This kind of features will give us no information to help classify our samples. So go back and and before splitting test and training sets, calculate the variance over each feature of your dataset and eliminate the ones with variance less than 0.01.

9. Now that you have fixed the singular matrix problem, repeat items 5 and 6.

10. Report accuracy, precision, recall, f1-score and the confusion matrix on both training and test sets for both LDA and QDA models. Which classifier performs better on mushroom dataset?

11. According to your confusion matrices, how many samples are misclassified in each classifier? Are these misclassified samples as FP (False Positive) or FN (False Negative)? Which one is more important in the task of classifying poisonous mushrooms?

12. And finally report the model parameters for both classifiers. How many parameters are there? How many parameters would you have if you used logistic regression?

4 Bonus

4.1 The Naive Bayes Algorithm

You will get the bonus score for implementing the Naive Bayes classifier on the Mushroom dataset from scratch. Therefore:

1. Separate your dataset into feature (X) and labels(y). Here X will be your feature matrix.
2. In order to evaluate you model, separate 20 percent as test set and the rest as training set.
3. Now before implementing your model, what is the difference between Naive Bayes classifier and LDA?
4. What are the model parameters of a Naive Bayes classifier? What is the role of each parameter? Between Naive Bayes, QDA, and LDA, which one has potentially more parameters? why?
5. Now that you are familiar with the parameters and differences, implement Naive Bayes from scratch. (Meaning learn the model parameters from training samples)
6. Then classify the test and training sets. Report accuracy, precision, recall, f1-score and the confusion matrix on both training and test sets. Amog the three Bayesian classifiers you implemented so far, which one performs better on mushroom dataset?
7. According to your confusion matrix, how many samples are misclassified? Are these misclassified samples as FP (False Positive) or FN (False Negative)?
8. And finally report the model parameters. How many are they? Compare the number of model parameters between three classifiers: Naive Bayes, LDA, and QDA.