University of Waterloo

ECE 657A:
Data and Knowledge Modeling and Analysis

Winter 2021

Assignment 2:

Classification using Naive Bayes, decision tree,
random forest, XGBoost
Parameter Estimation using MLE and MAP
Due: February 26, 2021 at 11:59pm EST

Overview

Collaboration: You may do your work individually or in pairs. You may collaborate with other students in the class on the right tools to use and setting up your programming environment but work on your own solution but be done by members of your group alone. Both members of the pair must sign up for a PairGroup in LEARN *and* in Crowdmark.

Hand in: One report per person or pair, via the CROWDMARK website in PDF, or image, format. You will need to divide the PDF up into one file for each `[CM#]` question. These files can be multiple pages. Some "questions" in this assignment have no output so no pdf if needed. It is best to start each of `[CM#]` solution on a new page and then drag and drop each onto the relevant question in Crowdmark. You should receive an invite to Crowdmark by email. Overlap and duplicated text between questions is alright, as long as the *entire answer* for each question fully contained within that question's pdf file. Also submit the code/scripts needed to reproduce your work as a python jupyter notebook to the LEARN dropbox.
Specific objectives:

- Load datasets and perform some exploratory plots.
- Study how to apply some of the methods discussed in class and gain experience on the use of classification algorithms: Naive Bayes, decision tree, random forest, XGBoost.
- Derive estimators for defined distributions using MLE and MAP.

Tools: You can use libraries available in python. You need to mention which libraries you are using, any blogs or papers you used to figure out how to carry out your calculations.

Dataset 1 - Wheat Seeds

This small dataset is the UCI Seeds Data set (`https://archive.ics.uci.edu/ml/datasets/seeds`). The examined group comprised kernels belonging

to three different varieties of wheat: Kama, Rosa and Canadian. These are your labels. There are 70 elements for each variety of wheat, randomly selected for the experiment. A python notebook will also be made available on learn to demonstrate how to load this dataset.

Dataset 2 - Covid-19 Outcomes in Ontario

Dataset 2 is about the confirmed COVID-19 cases in Ontario. We will provide a subset of this data on LEARN. The original data comes from the following file "Confirmed positive cases of COVID19 in Ontario" at :

`https://data.ontario.ca/dataset/confirmed-positive-cases-of-covid-19-in-ontario`

We will be using the features age group, gender, case acquisition info, city, outbreak, latitude, and longitude. You should convert categorical features to numerical or one-hot encoded feature, as appropriate. Take the feature `outcome1` as the label. Note, that in the full dataset on the Ontario website, the different classes are highly imbalanced so the dataset is quite challenging. We have sampled the data to be more balanced amongst the labels. While you can just download the whole dataset from the Ontario.ca website, for reporting results *you must use the dataset hosted on LEARN under Assignment 2.*

Question 1: Tree-based Classifiers and Ensembles

Part 1

- `[CM1]` : Dataset 1 - submit all the results for dataset 1 for these algorithms as one multipage pdf under Question 1 on Crowdmark.
- `[CM2]` : Dataset 2

Classify the data using three tree-based classifiers: Decision Trees, Random Forests and Gradient Tree Boosting. Tune the hyper-parameters of the classifier using 10-fold cross validation and sklearn functions. Evaluate the best value for the number of trees and maximum depth of trees.

For decision trees:

- max depth: {3, 5, 10, None}
- *None: (grow until leaf contains 2 datapoints)*

For this, plot the mean accuracy versus the maximum depth. Also, examine the final resulting splitting rules used for the trees. Do they indicate any interesting patterns that explain the data (particularly for the COVID dataset)?

For random forest:

- number of trees: {5, 10, 50, 150, 200}

- max depth: {3, 5, 10, None}

For this, the plot should be a **heat plot**. You should have (5 * 4) mean accuracies for different values of number of trees and maximum depth.

For Gradient Tree Boosting (on sklearn it is `GradientBoostingClassifier`):

- number of estimators: {5, 10, 50, 150, 200}

For this, plot the mean accuracy versus the number of estimators.
Note: the number of 'trees' grown by GBT is `n_classes` × `n_estimators` *but this is handled automatically.* Please leave the other parameters as default in sklearn.

Part 2

`[CM3]` **Analysis:** Across *both datasets*, compare and contrast the performance of the three approaches and point out any interesting patterns. Is the performance difference what you expected given the algorithm and the datasets?

Part 3 - Kaggle

Instructions will be posted on LEARN about a related Kaggle competition for this assignment once it is set up. Results from one of your ensemble methods on the Dataset 2 will be used.

Question 2: Naive Bayes Classifier

Part 1

- `[CM4]` : Dataset 1
- `[CM5]` : Dataset 2

Classify the data using the Naive Bayes Classifier. You can use GaussianNB in sklearn. Tune the hyper-parameters of the classifier using 10-fold cross validation and sklearn functions.

Evaluate the best value for the var_smoothing among the values {1e-10, 1e-9, 1e-5, 1e-3, 1e-1} and report the results using any performance measures you choose. Can you explain the impact of the smoothing parameter?

Part 2

`[CM6]` Considering *both datasets*, explain the performance of NB compared to the decision tree approaches in Question 1. Can you use the learned parameters of NB to make an interpretation about the data and compare this to the single Decision Tree model?

Question 3: MLE and MAP Derivation

In this problem we will find the *maximum likelihood estimator (MLE)* and *maximum a-posteriori (MAP)* estimator for given posterior and prior distributions. We will try to use MLE/MAP estimators to model the expected number of positive COVID-19 tests today. Specifically, we assume we have N samples, $x_1, \ldots, x_i, \ldots, x_N$ *independently* drawn from a *Poisson distribution*,

$$x_i \sim \text{Poisson}(\lambda) = \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}$$

X is a random variable that models a distribution over the number of people that will have positive COVID-19 tests on a given day i, in Ontario. The occurrence rate λ, is unknown. For MAP, we additionally assume a prior distribution for λ to be the *univariate normal distribution* with known mean and variance.

Note: *For this question you can submit your answer in any way which is clear to read and understand. If you can use LaTeX then create a new page, or document, for this question and submit it separately with your derivation typed out. You can also type math into* `Google Docs` *or* `Microsoft Word`*. If you submit a hand written solution, write very neatly and take a picture, or scan, of each page, and submit these images as your answer for the question.*

1. `[CM7]` Derive the MLE estimator for the occurrence rate λ. Make sure to show all of your work.
   Steps for MLE:
   (a) Derive the form of the likelihood term from the given posterior distribution:
   (b) Take the log of the likelihood
   (c) Take the derivative of the log-likelihood and set to equal to zero
   (d) Solve for the target parameter

2. [CM8] Now derive the MAP estimator for the infection rate λ. We assume that as a prior distribution that λ is itself sampled from a normal distribution[1] with known mean μ and known variance σ^2.

$$\lambda \sim \mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

Steps for MAP:

(a) Derive the form of the log-likelihood term from the given posterior distribution and the given prior distribution of it's parameter(s):

(b) Take the derivative of the log-likelihood and set to equal to zero. ***Note:*** *For this part, a clean analytical form for λ will not be possible. Derive the formula for MAP_λ in general terms using the known distribution's and reduce for λ as much as you can.*

Notes

You might find the following links are useful to solve this assignment:

- `https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html`
- `https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html`
- `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`
- `https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`
- `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html`
- `https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.GridSearchCV.html`

[1] Note: Since the rate λ cannot be negative, for correctness we would need to also assume $\mu > 0$ and is sufficiently far from zero, as well as ensuring λ is never sampled to be less than zero. Ignore this complication for the purposes of this question.