ML 4375 – Intro to Machine Learning

Homework 4 Overview

Worth 200 points

For this homework you will be implementing 2 machine learning algorithms in C++ and comparing the results and performance to the equivalent functions in R.

For this homework you can work with one other person or work alone if you prefer.

Steps:

1. (20 pts) Perform logistic regression on the given data set in an R script (not Rmd) using R library functions. Evaluate with the metrics indicated in details below. Your R script should also include at least 2 graphs and 4 R functions for data exploration.
2. (70 pts) Write a C++ program to implement logistic regression from scratch, and evaluate with the metrics indicated in details below.
3. (20 pts) Perform naive Bayes on the given data set in an R script (not Rmd) using R library functions. Evaluate with the metrics indicated in details below. Your R script should also include at least 2 graphs and 4 R functions for data exploration.
4. (70 pts) Write a C++ program to implement naive Bayes from scratch, and evaluate with the metrics indicated in details below.
5. (20 pts) Report. Write a summary of the accuracy and performance (run time) of the two approaches. Include screen shots of the R runs and the C++ runs for each algorithm. Cite references (any format) you used for the algorithm, including coding examples. Include screen shots of your R graphs. No particular format is required for either the report or references.

Turn in your 2 R scripts, 2 cpp files, data files, and report, zipped together.

Notes:

- Indicate in your summary how you computed run times. Here are some suggestions:
  - For the R scripts you can use proc.time() at the start and end of the machine learning part of the script and subtract the difference.
  - For the C++ programs, your IDE may give run time, otherwise measure from terminal.
  - Windows: https://stackoverflow.com/questions/673523/how-do-i-measure-execution-time-of-a-command-on-the-windows-command-line
  - Mac: https://stackoverflow.com/questions/26466572/mac-os-x-shell-script-measure-time-elapsed

Note: The timing for the R code should be only that portion running the algorithm, not parts that run data exploration functions or create graphs.

Details: Logistic Regression

- Data: plasma in library HSAUR. You will need to export it using write.csv() for your C++ program. Use all the data (32 observations) to build the model.
- R script:
  - train a logistic regression model on all the data, ESR~fibrinogen, using glm()
  - print the coefficients of the model
  - build the model "from scratch" in R as shown in the book
  - make sure you get the same coefficients in each approach
  - note that we are not doing test set evaluation on this data
- C++ program:
  - implement in C++ the same steps for logistic regression from scratch
  - feel free to use whatever data structures you like: arrays, vectors, etc.
  - if you have a linux system, you may want to check out the Armadillo library for matrix multiplication: http://arma.sourceforge.net/
  - feel free to use whatever programming paradigm you like, but make your C++ code fast

Details: Naïve Bayes

- Data: Titanic data set "titanic_project.csv" on Piazza. Use the first 900 observations for train, the rest for test.
- R script:
  - train a naïve Bayes model on the train data, survived~pclass+sex+age
  - print the model, which will show all the probabilities learned from the data
  - test on the test data
  - print metrics for accuracy, sensitivity, specificity
- C++ program:
  - implement naïve Bayes in C++; the code in the book should help
  - train/test on the same data as in the R script; output the same metrics
  - feel free to use whatever data structures you like: arrays, vectors, etc.
  - Here is a great video that gives a conceptual picture of naïve Bayes with Gaussian predictors: https://www.youtube.com/watch?v=r1in0YNetG8
  - The following formula shows how to calculate the likelihood of a continuous predictor. The book gives hints as well..

$$p(x = v \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

- Report
    - Write a summary of the two implementations, R and C++. Did you get the same results? How do the run times compare? How did you measure execution time?
    - Include screen shots of the output of each program
    - Include screen shots of the run times of each program
    - Write out the algorithm you used for training the classifier
    - Cite all references used
    - No required format for the report
- Be prepared to demo your code.