INFO371 Take-Home Final Exam/Winter 2018
15 pts/10% of the final grade

Your name:

March 7, 2018

Introduction

This is something in-between a lab and an exam. The form resembles a lab and you can:

- can use all available materials, including internet sources
- not communicate with the others. This is an individual exam.
- you can ask me if you are stuck with something. I'd like you to spend most of the time on ML problems, not on some technical misunderstanding. However, it's somewhat limited how much help I can offer.

The exam contains 3 questions. Each one is using a dataset you are familiar with and I do not ask you to explore the data. The first two just ask you to apply existing packages and functions, the third question is to implementing Naive Bayes. I try to do the tasks as simple as possible, so let's mostly ignore testing/training such.

As always, please submit your results in two ways: as a code (notebook/markdown/...) and as a final version (html/pdf/...).

Deadline: Today, March 7th, midnight.

Good luck!

1 Clustering (5 pt)

Your first task is to use k-means clustering to classify iris species. We'll use the iris data, perhaps the most widely used dataset in statistics.

Note 1: I evaluate here not how closely you can detect actual species, but instead how well you can do k-means clustering.

Note 2: You may want to consult the clustering notebook I demonstrated in class, see canvas files-in-class.

1. Load iris data. It is included both in R and in python (in sklearn).

For now, let's pretend we have no pre-knowledge about the actual species (after all, we do unsupervised learning) and we try to see if we can use k-means to classify the data.

2. Use only the petal/sepal data and perform k-means clustering. I expect you to use existing packages (but feel free to implement your own if you wish). Choose $k = 2, 3, 4$. Make a number of plots where you mark the cluster membership. Note: mark the cluster membership, not the species.

   Note: you'd like to do a 4D scatterplot but as this is not possible, you may either do a few 3D plots if you can, or just a few ordinary 2D plots.

3. Based on your plots, choose a single best clustering (no formal computations needed). Let this be your prediction.
4. Compare your prediction with the true species information. Did the clusters come close to the true species?

2 Regularization (5pt)

Here your task will be to estimate Boston house prices (medv) using all other features as predictors. Use ridge regression to avoid overfitting, and (10-fold) cross-validate the optimal regularization parameter.

I expect you to use existing packages and functions, not to implement the methods yourself. However, feel free implement your own, or to re-use something you have done for a previous problem set.

1. Load Boston data. It's located in R packages MASS, and in sklearn.
2. Select a range of regularization parameters λ. It should span a wide range of values from very small to very large, and contain at least 10 values in this range.
3. For each λ, use a linear ridge regression and compute the mean squared error (MSE) by 10-fold cross validation
4. Show a table (or graph) of λ and the corresponding MSE-s. Which λ performs best?

3 Naive Bayes (10pt)

Your first task is to take the good old 1984 house voting data, and based on votes, predict who is democrat, who republican. However, you have to implement Naive Bayes yourself. I recommend to consult Schutt & O'Neill (2013), chapter 4 and Daume, chapter 9.3, although just lecture slides should do as well.

Note: you may ignore smoothing here as we have very few features (16).

1. Read in the data (canvas-files-house votes). Ensure you know it's structure.

For Naive Bayes you need a number of frequencies:

- Pr(republican)
- Pr(democrat)
- Pr(voted yes on issue i|republican) for $i = 1 \dots 16$
- Pr(voted yes on issue i|democrat) for $i = 1 \dots 16$
- Pr(voted no on issue i|republican) for $i = 1 \dots 16$
- Pr(voted no on issue i|democrat) for $i = 1 \dots 16$

The data contains yeas, nays, and missings, your may just ignore the missings below.

2. Compute these probabilities.

   Hint: I recommend to create a vector for each of the latter 4 probabilities. Pr(D) and Pr(R) are just simple numbers.

Next, let's start implementing Naive Bayes. First we include just prior, and thereafter start adding voting data, vote-by-vote. We compute both probabilities–being democrat and being republican–and finally pick the larger value for each representative.

Note: instead of valid probabilities, I recommend to ignore the normalizing term and operate with log probabilities.

3. create prediction of being democrat and being republican for each representative based just on the prior.

   Note: if you operate with valid probabilities, obviously $\Pr(\mathsf{D}) = 1 - \Pr(\mathsf{R})$. However, this is not the case if you ignore the normalizer and compute in logarithms.

4. Now update the information, vote by vote: for each vote take the corresponding conditional probability, and update the Bayes estimate accordingly.

5. After including information about all 16 votes, you have your prediction. Print the confusion matrix, and calculate accuracy, precision and recall.

Congrats! You are done!