Homework 8

36-462/662, Data Mining

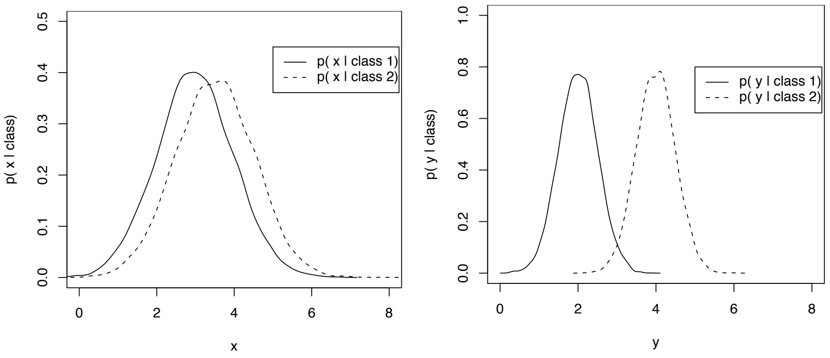
Due at 10 pm on Wednesday, 23 October 2019

Agenda: Thinking about information.

Important: The last two problems are much easier if you use the code accompanying lecture 15. You will find that code much easier to follow if you read the accompanying handout, on the class website, and not just the code.

There is no online quiz this week.

1. (10) Consider classifying images using their bag-of-colors representations. Each image has a count for each color. There are two classes. Our two favorite colors are x and y; the figures below show the distribution of pixel-counts for x (on the left) and y (on the right), with solid or dashed lines indicating the different classes.



   Which color gives us more information about the image's class? Why?

2. (10) Explain how a feature can provide information that lets us discriminate between classes, even though it has the same average value in each class. (You may want to draw some histograms.)

3. (10) Consider the following cross-tabulation table between two discrete variables, X and Y.

	$Y = +1$	$Y = -1$
$X = 1$	13	13
$X = 2$	90	10
$X = 3$	42	42
$X = 4$	72	8
$X = 5$	12	12
$X = 6$	117	13

There is (at least) one function b of X such that $I[Y; b(X)] = I[Y; X]$ but $H[b(X)] < H[X]$. That is, there is a way to compress X which loses no predictive information about Y. Find such a function b. Can you describe it in words?

4. There are two types of widgets, foos and bars. Some widgets contain baz and some do not. Consider the following contingency table.

	baz	
type	absent	present
foo	7	611
bar	250	694

   (a) (3) How many widgets are foos? How many are bars? How many contain baz? What is the probability that a random widget is foo? What is the probability that a random widget contains baz?

   (b) (5) What is the entropy of widget type?

   (c) (5) What is the entropy of widget type conditional on whether or not baz is present?

   (d) (5) What is the mutual information between widget type and whether or not baz is present?

5. Refer to the handout for lecture 15, where the greedy feature-selection is used to pick the seven most informative words in the *Times* corpus.

   (a) (6) How much information does each of those words provide about the class, given the other six words?

   (b) (6) Which words (if any) have positive interactions with the other six, and which (if any) have negative interactions?

   (c) (4) Use the prototype method and 1-nearest-neighbor classification on *all* the features; what is the accuracy, under leave-one-out cross-validation?

   (d) (6) What are the accuracies of the prototype method and the nearest-neighbor classifier using just the seven selected features? (Again, report results under leave-one-out CV.)

6. The code for lecture 15 contains a function to pick the q most informative features in a data-frame by greedy search.

(a) (5) How much information does the trio of words ("art", "painting", "evening") give about the story class?

(b) (10) Write a function which will pick the q most informative features to add to a given set of starting features. The function should take as inputs: a data frame, the column in the data frame which is to be predicted, the vector of features to start from, and q, the number of features to add. For full credit, the user should be able to specify features either by column numbers or column names.

Test your function by checking that it gives the same results as in Lecture 15 when started with "art" and ("art", "youre"), and $q = 1$ or $q = 2$.

(c) (5) What are the three most informative words to add to ("art", "painting", "evening")? How much information do they add? How much information do they provide on their own?

RUBRIC (10): The text is laid out cleanly, with clear divisions between problems and sub-problems. The writing itself is well-organized, free of grammatical and other mechanical errors, and easy to follow. Questions which ask for a plot or table are answered with both the figure itself and the command (or commands) use to make the plot. Plots are carefully labeled, with informative and legible titles, axis labels, and (if called for) sub-titles and legends; they are placed near the text of the corresponding problem. All quantitative and mathematical claims are supported by appropriate derivations, included in the text, or calculations in code. Numerical results are reported to appropriate precision. Code is properly integrated with a tool like R Markdown or knitr, and both the knitted file and the source file are submitted. The code is indented, commented, and uses meaningful names. All code is relevant; there are no dangling or useless commands. All parts of all problems are answered with actual coherent sentences, and raw computer code or output are only shown when explicitly asked for.