UWEO StatR201 – Winter 2013: Homework 5

Due: Friday March 8, 5 PM (grace period until March 14 lecture, with notification)

Assaf Oron, assaf@uw.edu

<u>Reading related to this assignment</u>: Lecture 8; Hastie-Tibshirani-Friedman, Sections 2.3, 2.4-2.5 (lightly), 9.2 and 13.3.

- Please submit online in the class dropbox. Please submit either ***.pdf is accepted as the main submission (with code pasted verbatim), or *.rmd.** (tip: to save time when using knitr, use `'cache=TRUE'` in the chunk header for any code chunks that run big simulations – this will prevent them from re-running each time you recompile the code).
- **Starred (*) questions and question-parts are not required.** You may submit them if you choose, or do any part of them without submitting.
- **Grading is determined chiefly by effort, not by correctness. If your submission shows evidence of independent, honest effort commensurate with the amount of homework assigned – you will receive full credit.**

1. Download the 'seedTrain.csv' dataset. It contains morphometric measurements of the kernels (=seeds) from 3 varieties of wheat grown in eastern Poland, 70 of each variety.

Most features are self-explanatory. "Compact" is a compactness measure, roughly speaking: how close the kernel is to being a sphere. "Groove" is the length of the groove that runs along the kernel. I assume all units (where applicable) are in mm.

The training set has only 50 samples of each variety.

Do some **basic** descriptives, such as `` `pairsPlus` ``, to make sure the dataset is intact and has no pathological issues. **Whatever happens, do NOT exclude data points.**

2. a. Run "`knn.cv`" on the dataset, as is (the off-the-shelf leave-1-out CV). Examine *k* between 1 and 15 in steps of 2. Decide on the optimal *k* (if there's a tie, choose the more parsimonious value).

b*. Examine the points missed by "`knn.cv`", and compare to the dataset's covariates. Consider whether a transformation and/or scaling and/or interaction of covariates might improve KNN performance, and try them out (that is: do whatever you think might work, and re-run "`knn.cv`").

c*. Write a general KNN CV function similar to "`rpartCV`", and do a 10-fold CV. Is the optimum at the same place as with the leave-1-out case?

3. a. Run CART via "`rpartCV`", the function on the class website, doing a **stratified** 10-fold CV (i.e., exactly 5 samples of each class in each CV group). Use `` `cp` `` as the tuning parameter, and scale it logarithmically as done in class. Examine at least 7 values, with the default (`cp=0.01`) in the middle.

If the optimum seems to lie outside your original range, expand it until you get a clear optimum. **Be sure to use `method="class"`, since the Variety variable is given as the numbers 1,2,3.**

b*. Examine the points missed by "`rpartCV`", and see whether things can be improved by, e.g., coding an interaction between some covariates and adding the product between two (scaled) covariates, as a new covariate (`` `rpart` `` **cannot** work directly with interactions in the formula, so you have to work around this limitation). **Remember, transforming covariates doesn't really matter for tree classification methods.**

4. When the test set is uploaded, download it and classify. Show the "confusion matrix" (that is, true classes in rows, vs. classifications in columns) for all methods you used (as a miminum, the standard '`knn`' and `'rpart'`).