Intro to Data Analytics and Visualizations

Lecture 36 – Intro to Support Vector Machines with Scikit-Learn
Fall 2014, November 17th

Outline

1. Support Vector Machines for Classification
2. Application with Python and Scikit-Learn
3. Inclass 12_1

Practice Problems

- Classify observations into two classes, just like we learned with the previous algorithms in R
- Validate

SVM

- Support Vector Machines (SVM) form a method that uses points in a transformed problem space that best separate classes into two groups.
- Classification for multiple classes is supported. SVM also supports regression by modeling the function with a minimum amount of allowable error.

Note*: See code for both R and Python implementations on different examples on Scholar/Resources/notes/Lect36.*

SVM(cont)

- Support Vector Machines (SVM) for classification use a function, called “kernel” to increase separation between classes by “transforming” data
- The “vectors” separate the “transformed” data into classes. The further apart the classes from each other, the better the separation and classification performs.
- Supervized learning

Note*: See code for both R and Python implementations on different examples on Scholar/Resources/notes/Lect36.*

Inclass Assignment 12_1

1) With R: calculate system time and AUC on KDDCup data (same data as for Inclass11, similarly just on training set). Do that for both a linear ('vanilladot') and gaussian ('rbfdot')kernels. Use and alter code in Lect36 folder.
2) Calculate AUC with Scikit learn and KDDCup data. You have to import data from R using pypeR first. Calculate AUC when using SVM for classification on both scaled and unscaled data. Do that changing the kernel as well to a linear kernel rather than rbf. See ipython notebooks on scholar (notes/lect36).
3) Use mtcars R dataset. Redo the classification from Inclass11 using SVM with gaussian kernel in both R and ScikitLearn. Calculate AUC. (you will have to import mtcars data in Scikit Learn. Use what you learned this semester)