Thesis Proposal

Elizabeth Ash

October 16, 2015

Research Question

Predictive modeling is a process used to create a statistical model of future behavior. A predictive model is made up of a number of **predictors**, which are variable factors that are likely to influence future behaviors or results. Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani give an example to briefly introduce the topic:

> Suppose we are statistical consultants hired by a company to provided advice on how to improve sales of a particular product... It is not possible for our client to directly increase the sales of the prodcut. On the other hand, they can control the advertising expenditure in each of the three media [TV, radio and newspapers]. Therefore, if we determine that there is an association between advertising and sales, then we can instruct our client to adjust advertising budgets, thereby indirectly increasing sales. In other words, our goal is to develop an accurate model that can be used to predict sales on the basis of the three media budgets (James, 2013).

In predictive modeling, data is collected for the relevant predictors, a statistical model is formulated, predictions are made, and the model is revised as additional data becomes available. My research project deals with predictive models and applications of such modeling.

An example of an application of predictive modeling is activity recognition. Activity recognition is an increasingly important technology because it can be applied to many real-life problems such as, home-based proactive and preventive healthcare applications. It can also be applied in learning environments, security systems, and a variety of human-computer interfaces. The goal of activity recognition is to recognize common human activities in real-life settings, such as weight lifting.

The article *Qualitative Activity Recognition of Weight Lifting Exercises* describes a study presented by Eduardo Velloso, Andreas Bulling, Hans Gellersen, Wallace Ugulino, and Hugo Fuks. Among other goals, the researchers wanted to provide feedback to weight lifters using qualitative activity recognition. The study involved six male subjects, all in their twenties and with little weight lifting experience. The subjects were taught how to lift a dumb-bell correctly and were also taught how to perform the same movement in four incorrect ways. The Unilateral Dumbbell Bicep Curl (see figure below) was the lift that was taught to the subjects. The five categories of lift data collected were:

* Class A: correct lift movement

* Class B: throwing the elbows to the front
* Class C: lifting the dumbbell only halfway
* Class D: lowering the dumbbell only halfway
* Class E: throwing the hips to the front

The subjects repeated each lift ten times and during each lift the researchers recorded a number of inertial measurements from sensors in the users' glove, armband, lumbar belt, and dumbbell (these are pieces of equipment that are commonly used by weight lifters). The sensors recorded several data points throughout the lifting motion and the final data set includes 160 variables.

Using this data, Dr. Homer White aimed to devise a random forest model to predict activity-type from those variables. Random forest is a predictive model method that is used for large data sets to find interactions between predictors. For any given set of values for a predictive variable, the random forest algorithm will return an estimate of the chances that it belongs to a certain class. For example, given a set of values for a certain variable from the Velloso, et. al data (let's say the roll_belt variable), then a random forest would return an estimate that the performed lift belonged to one of the five defined classes. The observation has a 5% chance of being in Class A, a 15% chance of being in Class B, 70% chance of being in Class C, 3% chance of being in Class D, and 7% chance of being in Class E. I would then conclude that based on the values, the lift is predicted to be a Class C error, meaning the subject is lifting the dumbbell only halfway up. (This is just a basic definition and hypothetical example, we will go into further detail in the Methodology section).

Dr. White designed a model that predicted the activity the subject was performing based on the data gathered during each of the lift types. The final model he constructed was estimated to be correct about 99.7% of the time. However, when the model was used to make predictions for new subjects, the results were terrible. The model would not be good to use to predict the activity of a new subject (White, 2015).

This is where my research project begins. The main questions I would like to investigate using the data from Velloso, et. al are "Can we predict how someone is performing a weight lift motion based on the numeric information provided from instruments? Furthermore, how well does random forest work to predict new lifts on the same subjects? What about predicting for new subjects?" I want to tweak Dr. White's model to build an honest predictive model-one to use in real life with new subjects.

Significance of Project

Regular physical activity is one of the most important things that can be done for overall health. It can help control weight, lower risk for heart disease, strengthen bones and muscles, and increase chances of longer life. However, if the activity is performed incorrectly, there is a greater risk of injury, which is counterproductive. To benefit most from a fitness routine, the activity should be performed as accurately as possible. Some people can go to a gym and work with a certified trainer, but many people cannot or will not work with a personal trainer. These people may be doing the correct lift motion, but there is no way to really know unless they are taught the correct motion by a professional.

In the case of my research project, I want to see if a predictive model can be made to recognize certain weight lift motions. If an honest predictor model could be made, then the model could be integrated into the weight lift equipment and used to determine if the lift was done correctly or incorrectly. This model could be integrated with other technologies and be used to help reinforce the correct weight lift motion by commending the user for a correct movement or making a comment when the user made an incorrect movement. For example, I am trying to perform the lift motion from the study correctly, but I am actually performing a Class C error. If my predictive model is good enough (based on the measurements from the sensors, the model can accurately predict in which class my lift belongs), then my armband could beep, notifying me of my error.

Methodology

The main goal of this project is to build a predictive model using data gathered from inertial measurement units (IMUs) of the Velloso, et. al study. To do so requires several steps of data cleaning, separation, and analyses.

General Considerations of Model Making

A new model should be able to be used to classify new data. Thus, it is important to have high model performance with new data. The performance of a model is measured in terms of its *error rate*: percentage of incorrectly classified instances in the data set (Witten and Eibe, 2000). The simplest way to get a handle on the ability of a predictive model to perform on future data is to try to simulate it.

The data set I am working with will have to be divided into two sets (a training set and a test set). The training set is used to build the model and the test set is new data that is used to measure the model's performance by being treated as new data. The model made with the training data will be tried out on the "new" test data. When the original data set is separated into the training and test sets, the simplest partition is a two-way random partition, careful to avoid introducing any systematic differences. The reasoning behind this type of division is that the data available for analytics fairly represents the real-world processes and that those processes are expected to remain stable over time (Steinberg, 2014). So, a well-constructed model will perform adequately on the new data.

You may be thinking that I should use all the data from the Weight Lifting data set. Then more data will be available to make the model and the model will be more accurate, right? However, this is incorrect. The *resubstitution error* (error rate on the training set) is a bad predictor of performance on new data because the model was built to account for the training data. The best model for predicting is the dataset itself. So, if you take a given data instance and ask for it's classification, you can look that instance up in the dataset and report the correct result every time. You are asking the model to make predictions to data that it has "seen" before- data that were used to create the model. Thus, to really know if the model would be a good predictor of the weight lift motion, it must be measured on the test data set, not the training set.

Since there are six subjects in the study, a total of 300 lifts were performed and recorded (each subject did 10 repetitions of the 5 lifts). However, during **each** lift the IMU measurements were gathered using a sliding window approach with different lengths (from 0.5 to 2.5 seconds), with a 0.5 second overlap. This resulted in a large data set (over 19,000 observations); a single observation in the data set corresponds to a specific time window for a specific subject performing one of the specified lifts. In his report, Dr. White separated the original data set into training and test sets using a random partition (a typical separation). However, in my project I will attempt to further expand on the partition by ensuring that no single lift appears in both sets. In real life, each new person does a new lift and no lift is the same. While the error rate for the model will most likely be worse with new data, I want to make the model as honest as possible, which means none of the measurements from a single lift should be in both the training and test set.

Types of Predictive Models

As mentioned earlier, I will be using random forest to build a predictive model. Another method that is useful for constructing predictive models is k-nearest neighbors (referred to as k-NN). Random forests are more sophisticated and this will be my main method of model building.

Random Forests

Random forest is a method for classification, regression, and other tasks, that operate by constructing a multitude of decision trees using a training set of data and outputting the class that is the mode of the classes (for classification) or mean prediction (for regression) of individual trees (Breiman, 2001). According to Leo Breiman (who helped develop the random forests technique), random forests are grown from many classification trees. It is a statistical algorithm that is used to cluster points of data in functional groups. When the data set is large and/or there are many variables it becomes difficult to cluster the data because not all variables can be taken into account. Therefore the algorithm can also give a certain chance that a data point belongs in a certain group (Breiman, 2007).

Disclaimer: The following part of this section becomes quite technical in regards to the forest growing process. Any non-technical reader may want to skip down to the next section

Each tree of the forest is grown as follows:

1. If the number of cases in the training set is N, sample N cases at random - but with replacement, from the original data. This sample will be the training set for growing the tree.
2. If there are M input variables, a number m<<M (m is considerably less than M) is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing.
3. Each tree is grown to the largest extent possible. There is no pruning.

The trees, which make up the forest, are then used to make the predictive model. I will be using R and R Studio coding to perfrom the random forests algorithm and use it to make a model.

k-Nearest Neighbor

The nearest neighbors approach, or better known as k-Nearest Neighbors algorithm (referred to as k-NN), is a non-parametric method used for classification. The input consists of the k closest training examples. In k-NN classification, the output is a class membership. An object is classified by a "majority vote" of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor (Mitchell, 1997). Basically, k-NN works based on minimum distance from what we are predicting to the training samples to determine the k nearest neighbors. After gathering the k-nearest neighbors, the majority of the neighbors are taken to be the prediction of the object. This is a simple method, but the results can be extremely helpful in determining where to begin for the random forest method.

Tools Used

For this project I am using a variety of tools which include R, RStudio, Git, and GitHub.

Expansion on R-related Tools

R is a programming language and an open source statisical program software environment used for statistical computing. R provides a wide variety of statistical and graphical techniques. R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License. It compiles and runs on a variety of systems, such as UNIX, Windows, and MacOS. To learn more about R and its contributors, please visit: https://www.r-project.org/.

R Studio is a free and open source integrated development environment (IDE) for R. This is a programmer's tool to help make writing and formatting code an easier task. To learn more about RStudio, please visit: https://www.rstudio.com/.

I am using R Markdown, which is an authoring format that enables easy creation of dynamic documents, presentations, and reports from R. This document format is extremely useful for

a statistical paper (which is what I will be working on this year), as it allows me to enter code chunks that are automatically executed as well as add figures, graphs, and charts. This format also allows me to save PDF, HTML, and Word versions of my document. However, this format is not very flexible when it comes to the aesthetics of the document. I cannot easily format my Bibliography nor do I know how to indent paragraphs because I am just beginning to use RMarkdown. If you require certain format for this paper, please understand that it will be very difficult with RMarkdown and I will probably not be able to focus on that until much later. To learn more about R Markdown, please visit: http://rmarkdown.rstudio.com/.

Also, my Thesis proposal is published on RPubs, which allows me to publish my documents to the web. You can go to the following URL to find a copy of my proposal: http://rpubs.com/esash77/thesisproposal

Expansion on Git and GitHub

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. To learn more about Git, please visit: https://git-scm.com/.

GitHub is a web-based Git repository hosting service. It offers all of the distributed revision control and source code management (SCM) functionality of Git. Unlike Git, which is strictly a command-line tool, GitHub provides a web-based graphical interface and desktop. It also provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project. GitHub offers both plans for private repositories and free accounts, which are usually used to host open-source software projects. To learn more about GitHub, please visit: https://github.com/.

My Thesis Proposal is loaded onto my GitHub account. On this site, you can write any comments by making an issue on my document. However, that requires you get a GitHub account (it is free!). Go to https://github.com/ and sign up for an account. You will just need to enter an email address and password to create the account. Once you are logged into your account, go to https://github.com/esash77/WeightLifting (which is my repository) and you can look at my proposal and the other files from Dr. White's initial model. On the right side of the screen there is a tab that is labeled as "Issues". This is where you would go if you wanted to make any comments about my proposal (or eventually my actual Thesis document).

Bibliography

Breiman, Leo, and Cutler, Adele. *Random Forests Leo Breiman and Adele Cutler.* Random Forests. 29 June 2007. Web. 20 Sept. 2015. https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#intro

Breiman, Leo. *Random Forests.* Machine Learning 45.1 (2001): 5-32. Web. 20 Sept. 2015.

James, Gareth, Witten,Daniela, Hastie, Trevor, and Tibshirani, Robert. *An Introduction to Statistical Learning with Applications in R.* 2013. Print.

Mitchell, Tom. M. 1997. *Machine Learning.* New York: McGraw-Hill.

Steinberg, Dan. *Why Data Scientists Split Data into Train and Test.* Why Data Scientists Split Data into Train and Test. 3 Mar. 2014. Web. 27 Sept. 2015.

Velloso, Eduardo, Bulling, Andreas, Gellersen, Hans, Ugulino, Wallace, and Fuks, Hugo. *Qualitative Activity Recognition of Weight Lifting Exercises.* Proceedings of the 4th Augmented Human International Conference on - AH '13 (2015).

White, Homer. *Predicting Movement-Types: Quick Model-Making with Random Forests.* Https://github.com/homerhanumat/WeightLifting. 4 Aug. 2015. Web. 1 Sept. 2015.

Witten, Ian H. and Eibe, Frank. 2000. *Data Mining: Practical Machine Learing Tools and Techniques with Java Implementations.* San Diego, CA: Morgan Kaufmann.

Xie, Yihui. *Dynamic Documents with R and knitr.* 2014. Print.

Working Outline

1. Introduction
    - Background information on project data (who did experiment, why, what is important)
    - Background information on the various approaches of model making
    - Significance of this project
2. Methodology
    - Brief summary of what was done to make the predictive model
    - Include the steps taken to create training/test data sets
    - Include the steps taken to create the model
    - Did I look at multiple model approaches?
3. Results
    - Data processing
    - Building the model
    - Estimate error rates
4. Conclusion/Discussiion
    - Discussion on how well models will work based on what was found in results
    - Any further investigations or applications?

Timeline for Completing Project

- Oct 1st: Complete first draft of Thesis Proposal
- Oct. 16th: Complete final draft of Thesis Proposal and send to Dr. Burch
- Learn background material by end of Fall Semester
    - Coursera courses

  - Meet with Dr. White

- By Spring Break 2016- finish making predictive models
- mid-April: First draft to Dr. White and Dr. Bowman
- Continue to work on drafts
- May 9th: Final draft due to Dr. Burch