CS 7641 Machine Learning
Assignment 1

David Yun

September 23, 2018

Abstract

The implementation, and basic fundamentals behind the following Machine Learning Algorithms will be discussed in detail:

1. Decision trees with some form of pruning
2. Neural Networks
3. Boosting
4. Support Vector Machines (SVM)
5. k-Nearest Neighbors (kNN)

For this assignment, Python is used, along with the necessary libraries associated within each topic. The full code files are available on github[1] Please refer to the README.txt file for concise instructions on how to run the files associated with each aforementioned algorithms. The full README.md file will guide a reader on how to use the files to better understand the objectives of this Assignment.

Contents

[1] David Yun's Github: https://github.com/tree-fiddy/Assignment_1

1 Decision Trees

The implementation of Decision Trees was borrowed from my previous coursework in CSE6242 (Data Visualization) with minor tweaks.

1.1 Dataset and Code

The dataset provided in this directory is the UCI Credit Approval Dataset[2]. Some small adjustments were made to the dataset, such as removing rows where data was missing.

1.2 Analysis

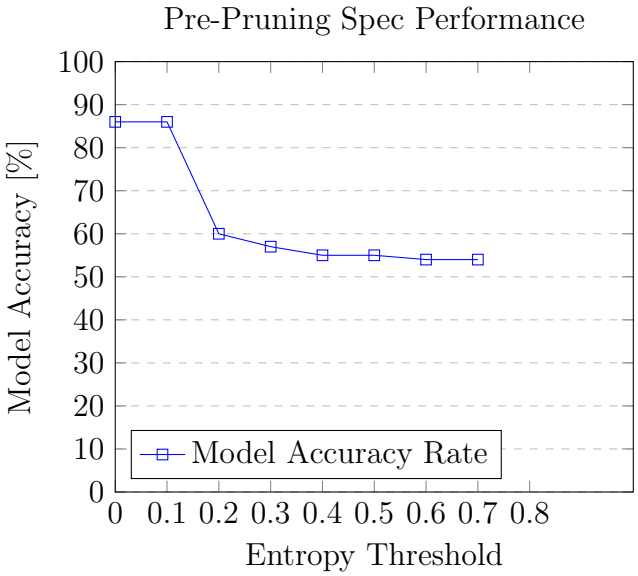
The accuracy of our decision tree relies on our tolerance for pre-pruning. In essence, one must specify *when* to stop growing the tree. On one extreme, we can set a condition to stop growing the tree once we reach a *pure*[3] leaf. However, this will likely make the tree longer than need be. A more depth-restrictive approach is to set an arbitrary threshold for Entropy in our Decision Tree, where we will halt further node and leaf generation if the current node produces leaves with Entropy values less than our specified threshold. The performance of our decision tree, while varying threshold levels is highlighted in Table 1.

Table 1: Pre-Pruning Spec Performance (forest size = 50)

Entropy Threshold	Model Accuracy
0.0	0.86
0.1	0.86
0.2	0.60
0.3	0.57
0.4	0.55
0.5	0.55
0.6	0.54
0.7	0.54

[2]UCI Credit Approval Data Set: http://archive.ics.uci.edu/ml/datasets/credit+approval

[3]A leaf is said to be "pure" if the leaf contains homogenous labels. By extension, this equates to a situation where Entropy equals 0.



As you can see, the model's accuracy reached an asymptote of around 50% accuracy, which indicates that setting a higher threshold for Entropy renders this model's efficacy no better than a coin flip. Thus, setting a pre-pruning specification is quite important here. More notable is the fact that using the extreme case where Entropy must equal 0 before considering the tree complete yielded the same results as a 0.1 threshold. However, computation times were slightly longer for this case, as more nodes had to be constructed until the threshold was met. While in our dataset, it wasn't a huge factor, this is an important consideration one should make when building models on much larger data. When the efficacy/accuracy of a model isn't compromised, any measures to speed up performance is a welcomed consideration.

2 Dataset 2: Credit Bureau Data

This dataset includes the following:

- First & Last & Middle names
- Marital Status
- Sex
- Birth Year
- Whether they owned real estate
- Current City
- Email Domain
- List of Monthly Payment status over 5 years for various tradelines

Like the alumni magazine dataset, this dataset can also be used for clusting and/or classification. Again, by itself, it's not useful beyond labeling a particular customer among a range of financial well-being compared to his/her peers. Perhaps, this dataset can be solely used for Prospecting (ie. obtaining a list of potential customers, and sending marketing offers). However, we will keep this paper limited in scope to leveraging data to understand existing customers, rather than prospecting potential customers.

3 Dataset 3: Company Web Site Tracking Data

This dataset includes the following:

- Title
- First & Last & Middle (initial) names
- Credit Card Type
- Credit Card Number
- List of products purchased in the past, with date of purchase and ship-to-address
- Which web pages the person looked at
- How long the person spent on each page
- What the person clicked on each page
- Estimate of how long the user's eyes spent on each page viewed (through camera)

This dataset is more worthwhile for the Company. It gives direct actionable response variables (Products purchased, web page landing statistics, click through rates...etc.).

Given this dataset alone, the company can use *Graph Theory* to understand the path in which a set of customers take to final purchase in order to optimize the customer experience to decreases the abandonment rate of online shopping carts.

Shopping fatigue can exist even through online channels. The plethora of buying decisions is not immune to online retail, and time spent hopping around products can be physically exhausting. In a literal sense, batteries can be exhausted while shopping, increasing the likelihood that shopping carts will be abandoned. To combat this, our company can look at the paths our average customer takes when making purchases. Perhaps there are too many forms being filled out to finalize a purchase, aiding in shopping fatigue.

The real value in this dataset lies on cross-selling potential. Again, *Graph Theory* can be used to determine cliques of items that tend to be included in a customer's basket of goods. By first identifying these cliques of items, a recommendation engine can be leveraged to increase sales while a customer is on the check out page. Furthermore, the retailer can investigate why certain products are not purchased together. For example, if retailers see

Table 2: Alumni-Credit Data Merge

Alumni Magazine	Credit Bureau
First and Last Names	First and Last Names
Email Domain	Email Domain
Current City	Current City

Table 3: Alumni-Credit-Company Data Merge

Alumni-Credit Merged Data	Company Data
First and Last Names	First and Last Names
Current City	Ship-to-City

a spike in peanut butter sales, but not a corresponding increase in, say, jelly sales, perhaps the retailer is charging too much for jelly.

4 Datasets Combined

The value of the datasets can be maximized by integrating them to obtain insights that a single dataset won't provide. While there aren't unique identifiers that can easily link the 3 datasets, one can achieve a reasonable amount of success in the following manner:

Notice that the company collected data only has City, First and Last name as common fields to be merged on. This reduces the reliability of the full dataset. However, we can still make a reasonable guess at the correct linkages, and assume Name + City is a unique identifier for our purpose here.

The data made available to us is immediately useful for the purpose of segmenting our existing customers. Using kMeans clustering, we can create distinct groupings of our existing customers and gather insights on key variables likely to predict shopping behavior.

Moreover, the combined datasets can be used for *prospecting* purposes. It is likely that level of education, and specifically, institution attended, are predictors of purchasing behavior. In order to test that, our combined dataset will allow us to see if it's worth pursuing/expanding an education demographic to increase sales. There are too many combinations of variables that can be analyzed to ascertain clusters, so I won't go into it here.

Last but not least, we can apply optimization models to determine which demographic we *should* be targeting at a particular time of day, month, or year. For example, with our purchased datasets, we may have determined that wives without a college education, who are current on their mortgages, who have high credit card debt, are very active on our retailer's website during 2 P.M. We can serve ads heavily to this demographic at the right times. Furthermore, we can tell our advertising partners to serve the bulk of our ads across all platforms where it is likely that our potential customers are making buying decisions around the same time every day, or particular time in the month. Our optimization model's *objective function* will be to **Maximize sales to people who are likely to restrain from**

making purchases in the middle of the day.

Our *variables* will be:

- Level of Education dummy variables
- Real estate owner dummy variable
- Marital Status
- Gender
- Month dummy variables
- City
- "Lurker" indicator variable
- Frequent Visitor indicator variable
- Penchant to Purchase, but witholding indicator variable

In our problem, we want to constrain our optimization model to hone in our demographic by including the following constraints:

- Frequent Visitor
- Lurker (person spends 30 min each visit)
- Penchant to purchase (eyes gaze on a product for more than a minute)

Using these constraints, we are likely to pick up customers who really are making a conscientious effort *not* to buy whatever the customer is looking at.

This optimization model will determine the most optimum demographic to pursue the right demographic to target our advertisements to. Implicit in our optimization model is the assumption that we are selling luxury goods- items that the average consumer can do without. Our goal is to first identify which cohorts are likely to yearn for an item. Then, our next goal will be to find the perfect promotion strategy (15%, 20%, 30%...etc. discount, Buy 1 get 1 Free, Email abandoned cart reminders...etc.) to entice the customer to finally make a purchase.