A decision tree for predicting the income of individuals from U.S. Census data

Mark Patrick White
CSIT 599G Computational Data Mining
Hood College, Computer Science Department
Frederick, Maryland, USA
mpw6@hood.edu

Summary

U.S. Census data contains a wide variety of demographic information about the U.S populace-at-large, in addition to historical data dating back several decades. Given this large collection of data, it is possible to use typical data mining techniques to discover novel and interesting facts about segments and classes of the citizenry of the country. I propose to examine in this paper a relatively abbreviated subset of the general census survey from 1996, and to specifically apply decision tree classification techniques in an attempt to accurately predict the category of income for individual citizens given a set of defining attributes for that person.

Introduction

Given the large amount of census data collected every ten years by the United States Census Bureau, demographic information for the American population is readily available and highly detailed. Taking into consideration the depth of the demographic data available, certain patterns and trends in income, gender, and other identifying features of U.S. citizens should be observable under a certain level of categorical scrutiny. The intent of this project is to use typical data mining techniques (most likely classification with a decision tree algorithm) on a large collection of U.S. Census data in order to predict the income range of future census entries. The goal of the project is to make accurate predictions on whether a person makes either up to or more than $50,000 annually, based upon certain attributes associated with that person's census information.

The United States Census Bureau is a principal agency of the U.S. Federal Statistical System responsible for producing data about the American people and economy. The Census Bureau is situated within the U.S. Department of Commerce and its director is appointed by the President of the United States.

The primary mission of the Census Bureau is conducting the U.S. Census every ten years, which allocates the seats of the U.S. House of Representatives to the states based on their population.[1] In addition to the decennial census, the Census Bureau continually conducts dozens of other censuses and surveys, including the American Community Survey, the U.S. Economic Census, and the Current Population Survey.[1] Furthermore, economic and foreign trade indicators released by the federal government typically contain data produced by the Census Bureau. The various censuses and surveys conducted by the Census Bureau help allocate over $400 billion in federal funds every year and help states, local communities, and businesses make informed decisions. [2]

Background

In conclusion, from these results there are several observations which become apparent. The first observation is that prediction with decision trees is heavily dependent upon the dataset, with specific regards to the variety of data in the records as well as number, position, and variability of the attributes. In the context of determining entropy (and thus, information gain), manipulating these varible aspects of the data attributes can alter the reults of entropy calculations in a decision, therefore creating situations of attribute characteristic dependency in the final decision tree product – a result not lending itself to the reliability of the outcome.

Another observation of decision tree prediction is that the final predictions are heavily dependent upon the quality of the decision tree generation algorithm. In the case of this investigation, the decision tree generation algorithm was a generalized version of the ID3 decision tree algorithm without any sophistication in its design. Due to this lack of sophistication, there was no intelligent branch pruning or restructure of the resultant generated decision tree, which lead to outrageously oversized (and, due to its size, uncompilable) decision tree logic. In order to create decision trees of manageable size, the training datasets required limitations in size and breadth, which inevitably led to less accurate predictions, by a factor dependent upon the magnitude of the limitations.

The last important observation witnessed during the investigation of this classification technique was that despite the susceptibility due to dataset manipulation and the unsophisticated method of generating decision tree logic, ID3 decision tree classification is a fairly effective means of classifying and predicting data. Assuming that conditions are favorable for a balanced, concise decision tree structure as output, the ID3 algorithm can make predictions with accuracy better than random, but with the caveat that the accuracy of prediction is still dependent upon the quality of the input (specifically, the training) dataset.

Approach and Methodology

The approach used in this investigation will attempt show the predictive capabilities of ID3 decision trees. To this end, a sample of U.S. Census data will be collected from the UCI Machine Learning Repository, and then that sample will be processed ensure the fitness of the data records and the structure of the dataset itself. With the initial data scrubbed, portions of the data will be replicated across a number of data subsets in order to meet the necessities of the software.

The replicated datasets will include a roughly one-tenth subset of the full compliment of data records to be used as the training set with which to generate the decision tree. Also to be included will be a dataset replicating all trained attributes minus the class to be predicted using every data record. The last dataset will also use every data record, yet will only include the class to be predicted as its only field.

With the replicated datasets in place, the ID3 algorithm is utilized to generate the decision tree logic using the training data set to build the code for the decision tree. Once the tree has be created, the decision tree logic is manually inserted into the driver code in order to process the records in the test dataset. Once the decision tree is embedded, the driver program is run with the test dataset being compared line-by-line in code with the single-attribute (the class of data predicted, in this case) validation dataset, and the aggregated results of the comparison are reported by the main driver application.

The methodology employed here involved a variant of a decision tree algorithm, modified to accept several attributes in the classification attempt. The classification algorithm was also designed to determine the expected salary category from a set of data preprocessed to eliminate data records where over a third of the attributes used by the decision tree are missing.

The data set consists of over 30,000 instances of census data, each instance representing a single citizen census entry. This data was collected by Barry Becker (of Silicon Graphics' Data Mining and Visualization team) from the 1994 U.S. Census database, from which a clean subset of data was extracted (filtered for citizens over sixteen that work a non-zero amount of hours-per-week and make over $100 annually). The data was subsequently donated in 1996 to the UCI Machine Learning Repository, from which the data set was taken [2].

The dataset is comprised of fourteen attributes to represent each instance, plus the classifier value (the annual income category) to be used in training the algorithm. The attribute fields include personal attribute types like the age of the citizen, their education level and a numeric representation of that education level, the citizen's marital status, family relationship (Wife, Husband, Not-in-family, etc.), as well as their race, national origin, and sex. Also, there are work-related attributes like "work class" (whether they are privately-employed, a government employee, or self-employed), capital gain and loss, hours-per-week worked; as well as the income classification category (>50K, <=50K). The attributes are a mix of types: categorical with multiple levels, binary, and numerical.

Thus, in order for the dataset to function within the limitations of ID3 algorithm (in terms of continuous attributes), the fnlwgt, education-num, capital-gain, and capital-loss attributes were removed, and the hours-per-week and age attributes' data values converted into categories.

The initial training dataset size used in the decision tree generation step of the process included the top 3,500 records from the preprocessed input dataset. The test dataset used consisted only of the attributes in the preprocessed input dataset, while the dataset used to verify the predictions in the driver program consisted only of the "income_class" test class from the preprocessed input dataset. Both the test and validation datasets contained all of the records from the original input dataset, limited by the attribute or class fields specified respectively for each.

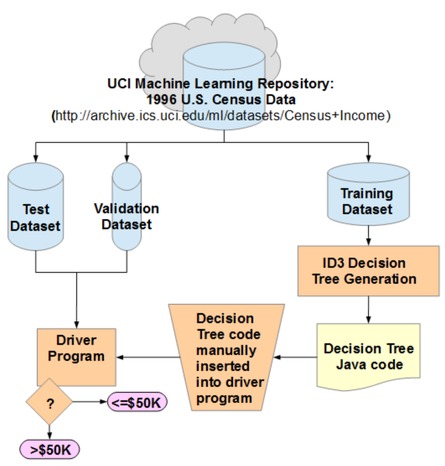

Figure 2: Process Workflow

Future Work

The results of this paper managed to present a remarkable volume of potential future options for investigation. The variability of the results based upon the configuration and subdivision of the input dataset revealed that further experiments preceded by manipulation of the data opens the realm of possibilities for additional study and experimentation to a nearly infinite degree of combination.

Future work may include (but is not limited to) the following areas of interest: larger subset of U.S. Census Bureau data, additional (or different) attributes included in the prediction, predictions of metrics other than income category, wider set of income categories, and changes to the income category ranges in the tested attribute class.

Additionally, alternative classification algorithms for generating decision trees could be used, or completely different means of classification (excluding decision trees). Finally, the particular ID3 decision could be streamlined to generate much smaller decision tree logic, allowing for more attributes to allowed in the prediction process.

Results

After implementing the initial design of the ID3 decision tree code and the predication driver program, a preliminary test of the decision tree code was conducted resulting in over one hundred thousand lines of decision tree logic (see Figure 4).

Unfortunately, when the resulting decision tree was embedded into the prediction driver program, the Java compiler was incapable of compiling the driver code, as the compiler can only handle up to 64K of source code in a single method; and specifically in the case of this tree, caused a buffer overrun (see Figure 5). This necessitated a series of sequential scale-back efforts in order to successfully provide a resulting decision tree.

The first attempt at scaling back the training input involved cutting back the number of training records used from 3,500 to 500 in an attempt to limit the number of possible categories for each attribute used in the prediction. The expected outcome was for there to be a smaller resultant decision tree with the drawback of having fewer categories of attributes taken into account in the prediction. A less accurate predication would also have been expected due to this consideration. In practice, the resulting decision tree had significantly fewer lines of code generated (in this case, approximately 35,000 lines of code), but still far beyond an acceptable size, in terms of what the compiler could handle.

Having generated a less-than-favorable outcome by limiting the number of records in the input dataset, it was decided that an approach involving limiting the number of attributes in order to reduce the complexity of the decision tree may offer better results. The strategy applied to this approach was to identify the attributes with the largest number of distinct categories and eliminate them from all of the input datasets used in the experiment. The attributes identified as having the largest number of data categories were education, occupation, and native country. Due to fears that eliminating all of these attributes would result in insignificant results and an overall poor-quality decision tree, only two of the attributes were removed from the dataset -- leaving the education attribute intact based upon experimenter bias.

Using the same 3,500 records as the initial decision tree generation attempt, this method generated a decision tree of roughly 7,500 lines of code – still too large for the compiler (see Figure 6). Despite the lack of success in compiling this decision tree code, this attempt demonstrated an impressively significant decrease in the number of lines of decision tree source code generated.

Based upon the results of these two previous attempts to limit decision tree code size, a combination of the two restrictive techniques was applied, which demonstrated success in compiling the code. In the end, a learning set of 2000 of the first records from the original dataset were used for the training dataset (using the narrowed composition of eight attributes instead of the initial ten), and it was determined to use an additional training set of only 500 of the first records from the original dataset (also using eight attributes) in order to identify any artifacts would arise from restricting the training dataset record size to a high degree.
These decision tree code sets were embedded into respective prediction driver programs are run to aggregate the overall results of each predicted record (out of the total 32,562 from the test and validation datasets). For the test dataset containing the decision tree generated from 2000 training records, the driver program successfully predicated 19,216 of the test records and failing 13,346 predictions: a success rate of roughly 59.0% (see Figure 7).

Surprisingly, when the test dataset containing the decision tree generated from 500 training records was used in the driver program, a total of 20,745 records were successfully predicated, with 11,817 predictions failed: a success rate of roughly 63.7% (see Figure 8).

The higher number of successful predictions by the prediction driver program using fewer learning records to generate the decision tree initially caused some confusion. However, it was determined that the particular ordering of the records in the training and test datasets contributed to situation allowing excessively low numbers of training records to generate a decision which had conditions that were more favorable to predicting a simple set of test records correctly but without much reliability. As the number of training records increase, an eventual drop in predictive success is conjectured to occur as the prediction results become more reliable, and steadily increasing in both predictive success and reliability of the predictions as the training set record size increases. Future work, however, would be necessary to investigate that conjecture.

Finally, the full original dataset was used in Weka (as both as the training dataset and test dataset) to get an idea of how successful the built-in implementation of the ID3 decision tree algorithm in Weka is at predicting income class results. Though not necessarily significant with respect to the algorithms and datasets used in this paper, the results were nevertheless interesting and showed that a quality implementation of the ID3 decision tree generating algorithm can still be fairly effective in predicating outcomes. As can be seen in Figure 9, Weka (using its built-in ID3 algorithm against the full set of 32,562 records) successfully predicated 28,033 of the test records and failing 4,528 predictions: an 86.1% success rate.

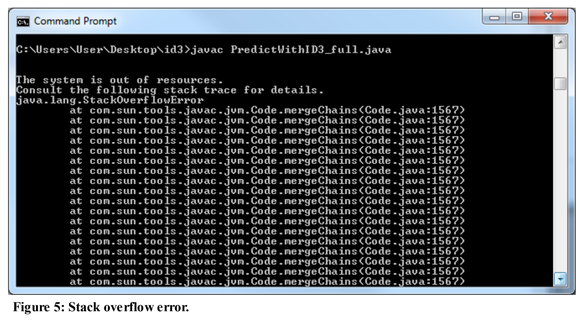

Figure 5: Stack overflow error.

Figure 6: Java compiler fails due to "code too large" error.

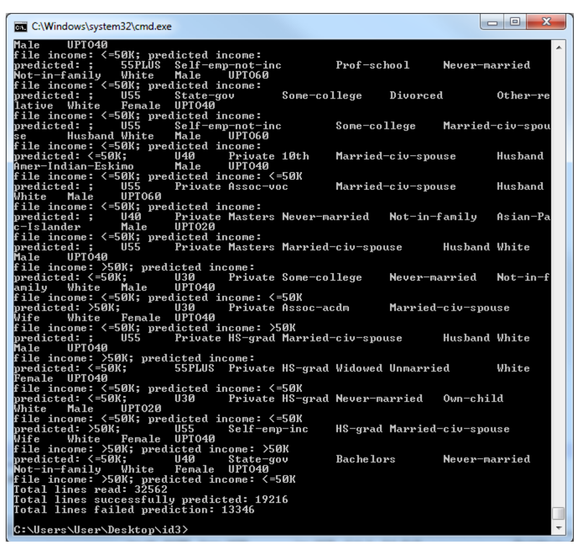

Figure 7: Trained with 2000 records.

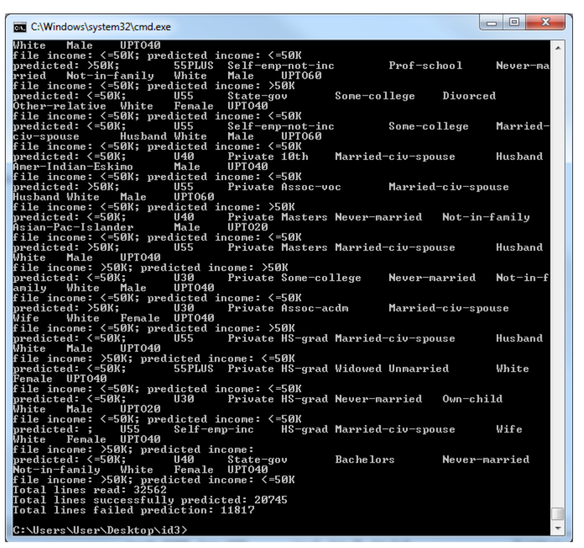

Figure 8: Trained with 500 records.

Conclusions

The results of this paper managed to present a remarkable volume of potential future options for investigation. The variability of the results based upon the configuration and subdivision of the input dataset revealed that further experiments preceded by manipulation of the data opens the realm of possibilities for additional study and experimentation to a nearly infinite degree of combination.

Future work may include (but is not limited to) the following areas of interest: larger subset of U.S. Census Bureau data, additional (or different) attributes included in the prediction, predictions of metrics other than income category, wider set of income categories, and changes to the income category ranges in the tested attribute class.

Additionally, alternative classification algorithms for generating decision trees could be used, or completely different means of classification (excluding decision trees). Finally, the particular ID3 decision could be streamlined to generate much smaller decision tree logic, allowing for more attributes to allowed in the prediction process.

References

[1] Saini, R., P. Singh. "Classification using decision trees," Internet: http://iasri.res.in/ebook/win_school_aa/notes/Decision_tree.pdf. [July 18, 2014].
[2] Bache, K., Lichman, M., "Census Income Data Set". Internet: https://archive.ics.uci.edu/ml/datasets/Census+Income. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science. [July 18, 2014].
[3] Antonov, A., "Classification and association rules for census income data", Internet: http://mathematicaforprediction.wordpress.com/2014/03/30/classification-and-association-rules-for-census-income-data/. Mathematica for prediction algorithms: Using Mathematica implementations of machine learning algorithms [http://mathematicaforprediction.wordpress.com]. (March 30, 2014).
[4] Wikipedia. "ID3 algorithm," Internet: http://en.wikipedia.org/wiki/ID3_algorithm. [July 18, 2014].
[5] Wikipedia. "United States Census," Internet: http://en.wikipedia.org/wiki/United_States_Census. [July 18, 2014].
[6] United States Census Bureau. "United States Census Bureau," Internet: http://www.census.gov/en.html. [July 18, 2014].
[7] Tan, P., Steinbach, M., Kumar, V. Introduction to Data Mining. Boston, MA: Pearson Education, Inc., 2006. [July 18, 2014].
[8] Bahety, A., "Extension and Evaluation of ID3 – Decision Tree Algorithm". University of Maryland, College Park. [July 18, 2014].
[9] Saini, P., Rai, S., Jain, A. K., "Decision Tree Algorithm Implementation Using Educational Data," International Journal of Computer-Aided technologies (IJCAx) Vol.1, No.4, April 2014.