HHS Public Access

Author manuscript

Proc IEEE Symp Comput Intell Healthc Ehealth. Author manuscript; available in PMC 2017 December 08.

Published in final edited form as:

Proc IEEE Symp Comput Intell Healthc Ehealth. 2014 December ; 2014: 187–190. doi:10.1109/CICARE.2014.7007853

FDT 2.0: Improving scalability of the fuzzy decision tree induction tool - integrating database storage

Erin-Elizabeth A. Durham,
Department of Computer Science, Georgia State University, Atlanta, USA

Xiaxia Yu, and
Department of Computer Science, Georgia State University, Atlanta, USA

Robert W. Harrison
Department of Computer Science, Georgia State University, Atlanta, USA

Abstract

Effective machine-learning handles large datasets efficiently. One key feature of handling large data is the use of databases such as MySQL. The freeware fuzzy decision tree induction tool, FDT, is a scalable supervised-classification software tool implementing fuzzy decision trees. It is based on an optimized fuzzy ID3 (FID3) algorithm. FDT 2.0 improves upon FDT 1.0 by bridging the gap between data science and data engineering: it combines a robust decisioning tool with data retention for future decisions, so that the tool does not need to be recalibrated from scratch every time a new decision is required. In this paper we briefly review the analytical capabilities of the freeware FDT tool and its major features and functionalities; examples of large biological datasets from HIV, microRNAs and sRNAs are included. This work shows how to integrate fuzzy decision algorithms with modern database technology. In addition, we show that integrating the fuzzy decision tree induction tool with database storage allows for optimal user satisfaction in today's Data Analytics world.

Keywords

fuzzy logic; fuzzy ID3; Big Data; HIV protease; drug resistance prediction

I. Introduction

Decisioning, or the machine emulation of human learning and classification, is a nebulous area in computer science. Classic decisioning problems can be solved given enough time and computational power, but discrete algorithms cannot easily solve fuzzy problems. Fuzzy decisioning can resolve more real-world fuzzy problems, but existing algorithms are often slow, cumbersome and unable to give responses within a reasonable timeframe to anything other than predetermined, smaller dataset problems. As the volume of data available for analysis grows in the modern world, it is becoming more and more imperative that effective machine-learning solutions are examined that can efficiently handle large datasets.

The effectiveness with which fuzzy decisions can be resolved via the Fuzzy Decision Tree algorithm is significantly improved when using a database as the storage unit for the fuzzy

ID3 objects, versus standard flat files and in-memory java objects. Furthermore, we demonstrate that pre-processing certain portions of the decisioning within the database layer can lead to better membership classifications, especially on large datasets. Large biological datasets from HIV, microRNAs and sRNAs were used to measure the effectiveness of the tool. microRNA and sRNA are sequence-based function prediction for RNA in eukaryotes and prokaryotes, respectively. The HIV data are prediction of drug resistance from protein sequence data. Additional datasets from the UCI Repository of Machine Learning Databases and from private industry were used to demonstrate the range of the new tool, from small (<150) to truly large (>400000).

The freeware fuzzy decision tree induction tool, FDT, uses an improved fuzzy ID3 (FID3) algorithm to perform its fast decisioning [1]. Fuzzy sets were coupled with the Quinlan ID3 partitioning algorithm to generate the decision trees that are the basis of the tool's logic [2]. As decision trees are notoriously sensitive to small changes in training data [3], and largely unable to cope well with uncertain/variable data, FDT 1.0's implementation of fuzzy sets and fuzzy reasoning used approximation to deal with the data set noise: uncertainty/inexact data and fluctuations in data precision, etc. The result was a to create a rigorous and effective decisioning tool [1].

FDT 1.0 is a java-based application which does not account for data retention for future decisions, and therefore needs to be recalibrated from scratch every time a decision is required. It outperforms C4.5 and the genetic algorithm tree on every dataset against which it was tested, and it outperforms Random forest on many [1].

FDT 2.0 captures the base data, the fuzzification model, and the decision information into a relational database, from which future decisions can be extrapolated.

FDT 2.0 brings a comparable accuracy level, with the added benefits of having the training/test sets maintained in a stand-alone database, each dataset now with its own identifiable set of database objects that can be dropped and reused (or maintained, as desired by the tool audience) independently, allowing multiple training/test sets to coexist in the tool without interfering with each other. Additionally, the tool users can independently explore the data afterwards via adhoc SQL querying in the resultant relational database, drawing further conclusions from the resultant data, and/or manually pruning data as desired from the training sets.

The rest of the paper is organized as follows: Section II introduces FDT 2.0: a database storage version of the fuzzy decision tree induction tool, including an overview of the new FDT 2.0 algorithm. Section III presents our experiments results and Section IV concludes the paper.

II. FDT 2.0: NEW AND IMPROVED!

A. FDT Algorithm

The FDT algorithm couples the Quinlan Iterative Dichotomiser 3 algorithm to recursively create decision trees, with a fuzzy data/fuzzy membership representation to deal with

uncertainty, noise, and outlier data elements that normally would cause the Quinlan ID3 algorithm to falter due to its sensitivity to small changes in training data [1].

There are 4 steps involved in the FDT fuzzy decision tree induction algorithm [1]:

1. Data Fuzzification.
2. Generating the fuzzy decision tree.
3. Converting the fuzzy decision tree into a set of fuzzy rules.
4. Inference.

Step 1 of the FDT fuzzy decision tree induction algorithm involves calculating the membership values of the supplied data, either using a fuzzy membership function supplied by the Domain Experts associated with the supplied data or automatically generated based on the contents of the data itself. Next, is the actual building of the fuzzy decision tree. The training data is recursively partitioned based on the values of an attribute chosen via an information theory measure. Multiple choices exist for the information theories measures and fuzzy membership functions. The fuzzy decision tree (FDT) is then boiled down into fuzzy rules of the form “if p then q.” The final step is inferring matches from the dataset to be tested, against the generated fuzzy rules.

FDT 1.0 is a java-based application that implements the Fuzzy Decision Tree (FDT) algorithm: the integration of the Quinlan ID3 decision-tree algorithm together with fuzzy set theory and fuzzy logic[1]. In existing research, the Fuzzy Decision Tree produced comparable results and/or outperformed other machine learning algorithms including Random Forest, C4.5, SVM and Knn [1], and is therefore a prime candidate for integration with a database to facilitate larger data set analysis.

B. FDT 2.0

To create FDT 2.0, we took the FDT algorithm and mapped it to relational database constructs, using the objects inherent to a database: separated schemas, indexing, partitioning, pipe-and-filter transformations, preprocessing data, materialized and regular views, etc. These database objects are already optimized for use in a database, and one separated the heavy-processing data-manipulation logic and placed it in the database layer with the data itself, with excellent results. Using the freeware MySQL 5.5 as the database software, the FDT-Database software performed very well on larger datasets, running into hundreds of thousands of records in the training and test data files.

Each training/test set now had its own identifiable set of database objects that could be dropped and reused (or maintained, as desired by the tool audience) independently, allowing multiple training/test sets to coexist in the tool without interfering with each other. Comparable accuracy results were achieved, and larger datasets could now be classified by the tool. As the training data is held in a database, it is not necessary to rerun the training steps whenever the application is invoked, as it is with the FDT 1.0 too1 [1]. There is no imposed limit to the number of test/train sets that can be stored, and accurate rulings can be added back into the training set (and training steps (b)-(d) can be re-run), to grow the “knowledge-base” of the training set.

C. FDT 2.0 Algorithm

Training/Learning Steps:

- **a.** Load the multi-variant training data into the training table.
- **b.** Run statistical analysis on the training data to determine the frequency, probability, entropy, dominant value, and other statistical measures for each attribute and category.
- **c.** Based on the training data, determine the if-then decision tree.
- **d.** Based on the training data, determine the fuzzy set membership qualifications.

Prediction/Testing Steps:

- **a.** Load the multi-variant test data into the test table.
- **b.** Predict the classification for the multi-variant test data.
  - **i.** Determine memberships
  - **ii.** Evaluate fuzzy rules
  - **iii.** Combine outputs of fuzzy rules
  - **iv.** Determine crisp classification prediction
- **c.** Measure the accuracy of the classification prediction.
- **d.** (optional) Add accurate prediction rulings back into training table, and rerun training.

III. EXPERIMENT

A. Solution Viability

The purpose of the experiments was to determine whether the Fuzzy Decision Tree algorithm could be implemented to take advantage of a database storage structure, in order to facilitate decisioning larger data sets. Initial comparisons were run using the testing databases from UCI Repository of Machine Learning Databases to make comparisons against the earlier implementations of the FDT that had not used a database storage structure. Large biological datasets from HIV, microRNAs and sRNAs were used to measure the effectiveness of the tool. Later experiments utilized classified large data sets from private industry.

B. Datasets

For the first set of experiments with the FDT classification tool, we used four (4) of the publically available datasets from the UCI machine learning repository [4]: Shuttle, microRNA, sRNA and Iris. This included the biological datasets for sequence-based function prediction for RNA in eukaryotes and prokaryotes, microRNA and sRNA, respectively. The datasets varied widely in type and attributes, from Shuttle with a set of 58000 multivariate items with 9 integer attributes and 7 possible classifications; to Iris, with a set of 150 multivariate items with 4 real attributes and 3 possible classifications.

The data sets were pre-processed into SVM-like training and test files. Then the datasets were run through the FDT classification tool, and the prediction results evaluated.

After completing these experiments satisfactorily, we went looking for larger data sets to classify. For the larger datasets, we used classified datasets of HIV-1 protease mutant structure/inhibitor complexes from the genotype-phenotype datasets at Stanford University [5]. The HIV data are classified prediction data of drug resistance from protein sequence data. The datasets contained approximately 10000 to 20000 classified records, including 211 varied (integer and decimal) attributes and 2 possible classifications, each. For the second set of even larger datasets, we are indebted to Momentum Telecom for supplying sample datasets of modem quality of service (QoS) DOCSIS[6] classification data. For the modem QoS datasets, the resultant measurements and evaluation data with each modem classified into a Red/Bad, Yellow/Warning, and Green/Good state was provided. The datasets included up to roughly 500000 multivariate items, each with 9 decimal attributes and 4 possible classifications.

C. Results

The initial iteration of the FDT classification tool had a very difficult time dealing with larger datasets. Run times were extremely long and memory errors abounded. For the same configuration settings and datasets, the original FDT 1.0 and the new FDT 2.0 classification tools work comparably (very well) on smaller datasets. The FDT 1.0 classification tool has a difficult time dealing with larger datasets in a timely fashion, and has no way of storing decisions or multiple dataset decisioning models. For larger datasets, FDT 2.0 brings a comparable accuracy level, with the added benefits of having all of the datasets maintained in a stand-alone database, each dataset now with its own identifiable set of database objects that can be dropped and reused (or maintained, as desired by the tool audience) independently, allowing multiple training/test sets to coexist in the tool.

IV. CONCLUSIONS AND FUTURE WORK

In this paper we presented a new implementation of the freeware fuzzy decision tree induction tool (FDT 2.0). FDT 2.0 implements a relational database backend to store the test base dataset(s), the fuzzification model(s), and the decision information, from which future decisions can be made without having to rerun the decisioning process. FDT 2.0 has a comparable accuracy level to FDT 1.0, with the added benefits of having the datasets maintained in a stand-alone database, each dataset now with its own identifiable set of database objects independently, allowing multiple datasets to coexist in the tool without interfering with each other. Additionally, the FDT 2.0 tool users can now independently explore the data afterwards via adhoc SQL querying in the resultant relational database, drawing further conclusions from the resultant data, and/or manually pruning data as desired from the datasets.

The optimization of fuzzy decisioning algorithms in order to approximate expert human judgment and disambiguate classifications is incredibly important for working towards the ability to efficiently work with larger datasets. These larger datasets include multivariate classifications from DNA to Climate Analysis. As the training data is held in a database, it is

not necessary to rerun the training steps whenever the application is invoked – with larger datasets, this can be a huge time savings. The FDT classification algorithm outperforms other machine learning algorithms including Random Forest, C4.5, SVM and Knn, and is therefore a prime candidate for integration with a database to facilitate large dataset analysis.

The fuzzy decision tree induction tool (FDT) is still under development. Future directions include increasing the accuracy of predictions, decreasing the speed of predictions, incorporating more automated “learning” elements and working towards decisioning models that can comfortably work with terabyte-sized data sets in a reasonable time frame.

Acknowledgments

This work was supported in part by the Georgia Cancer Coalition (RWH is a Georgia Cancer Scholar) and the Georgia State University Molecular Basis of Disease Initiative. This research was supported, in part, by the National Institute of Health grant U01-GM062920.

References

1. Abu-halaweh, NM., Harrison, RW. FDT 1.0: An improved fuzzy decision tree induction tool. Fuzzy Information Processing Society (NAFIPS), 2010 Annual Meeting of the North American; July 2010; p. 1-5.
2. Janikow CZ. Fuzzy Decision Trees: Issues and Methods. IEEE Trans on Man, Systems and Cybernetics. 1998; 28(1):1–14. [PubMed: 18255917]
3. Yuan Y, Shaw MJ. Induction of Fuzzy Decision Trees. Fuzzy Sets and Systems. 1995; 69(2):125–139.
4. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/
5. Genotype-Phenotype Datasets. Stanford University HIV Drug Resistance Database. Retrieved March 12, 2014, from http://hivdb.stanford.edu/cgi-bin/GenoPhenoDS.cgi
6. Data Over Cable Service Interface Specifications. Cable Labs. Retrieved Jan 02, 2014, from http://www.cablelabs.com/specifications/CM-SP-PHYv3.0-I08-090121.pdf

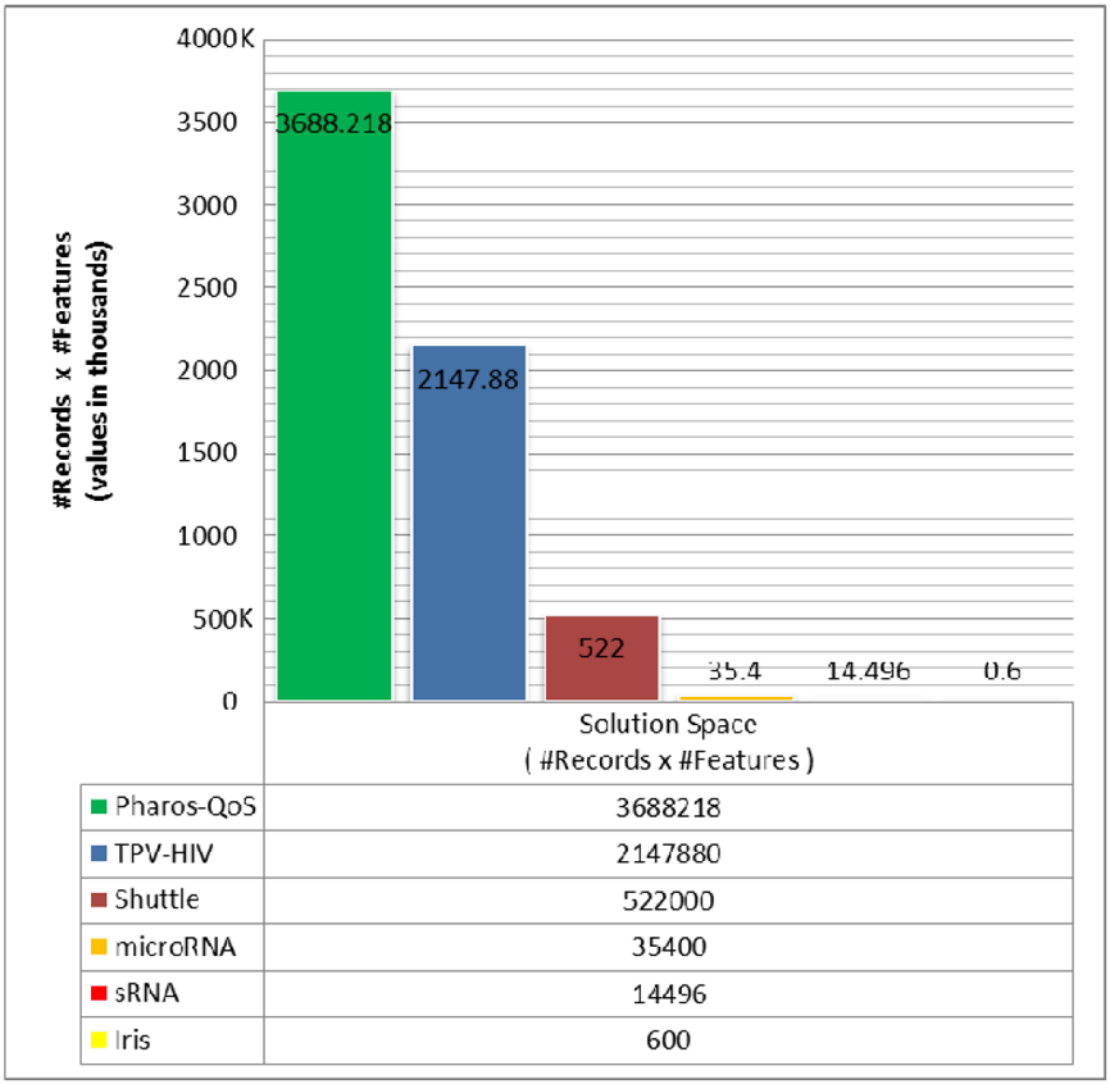

Figure 1.
Properties Of Each Dataset Solution Space Used During Experiments.

TABLE I

Properties Of The Datasets Used During Experiments.

Name	Source	Records	Features	Categories	Size
Pharos-QoS	Momentum Telecom	409802	9	4	XL
TPV-HIV	Stanford University	10228	210	2	L
Shutle	UCI machine learning repository	58000	9	7	L
microRNA	UCI machine learning repository	4425	8	2	M
sRNA	UCI machine learning repository	1812	8	2	M
Iris	UCI machine learning repository	150	4	3	S

TABLE II

FDT 1.0 Versus FDT 2.0 Accuracy Measures.

				FDT 1.0	FDT 2.0
Name	**Features**	**Categories**	**Size**	**Accuracy (%)**	**Accuracy (%)**
Pharos-QoS	9	4	XL	(too large)	96.5
TPV-HIV	210	2	L	(too large)	99.95
Shuttle	9	7	L	82.44	83.66
microRNA	8	2	M	83.09	82.46
sRNA	8	2	M	50	71.68
Iris	4	3	S	97.22	91.67