Brian DiZio

Artificial Intelligence

Dr. McCoy

October 9, 2016

**Extra Credit Project: Business Model Process AI**

**Group Members and Their Roles**

Brian Dizio [Coordinator, Documentor, Implementor]

**Overview**

This project mainly focuses on the identification and classification of the product, the crucial element to success, of the attached business model process patent application. I would like to implement tasks (product identification) of the business model into a computational agent for three purposes: to challenge human insight based on historical data analysis (assuming the patent is implemented), to provide a tangible item for the patent application, and to provide an intellectual exercise with which to consequently generate improvements and refinements for the entire business process model patent. Specifically, I would like to use offline machine learning to find the best product to sell using my business model process. This would include variables relevant to an environmental scan of the industry such as product bulk-sales, brand importance, product specifications, ease-of-inventorizing potential, economic factors, consumer trends, exclusion from other retailers and more. Further with the agent as the computer program, we have a knowledge base as follows:

where

is a function of time and the environmental scan variables mentioned above (product bulk-sales, etc).

**Completion Summary:**

The projected was completed to about 90% of readiness to simulate within three different programs, Excel, MATLAB, and Weka.

*Excel:*

Data was researched in American University business databases to gain some insight into what type of data is collected. Then, Demographic Matrices were constructed with variables that appear to be optimal for proving a consumer would buy a product in bulk, and that would be present in researched and available data. The Demographic Matrices help to distinguish which demographic the product would be best selected for. As of now, the heuristic is the sum of all the percentages, but it will certainly be molded before finalization. Next, data for each product is sourced from each demographic matrix for each year in the Product Matrices. This gives relative standing to which product would sell the most in bulk. Right now, it appears the best product would again have highest the sum of the percentages. This heuristic is not yet included in the data set but rather computed for insight while in the software development process. Also, a macro must be written that automatically copies the data from the Demographic Matrices into the Product Matrix for the current year.

Successes:

-Data research

-Conceptual creation of Demographic Matrices

-Conceptual creation of Product Matrices

-Unit test simulation

Necessary Improvements:

-Need more variables

-Need variables to exactly prove the product will be sold in bulk

-Need to make the heuristic reflect dollar amounts in revenue

-Need variables to actually be available and not idealized

-Need real market data

**-**Need simulation to have independent distributed variables. Right now they are dependent.

-Need macro to automatically copy data from Demographic Matrix to bottom of Product Matrices data.

*MATLAB*

After the Product Matrices have been computed, after data is filled in the Demographic Matrices for the current year (currently simulated data is used for all 20 years, the exact same data for each year as a unit test for the classifier), MATLAB imports the data from Excel in matrix format. A Principal Component Reduction is used, with whatever variance you would like explained, and the data is reduced to less variables. Again, the heuristic of sums of percentages for all variables is used as insight and the data is stored a format ready to be exported to an .arff file for Weka classification.

Successes:

-Principal Component Reduction code

-Data formatting and writing

Necessary Improvements:

-Do the reduced variables still represent percentages? Can the same heuristic be used of sums across variables for each product?

*Weka*

The data is outputted from MATLAB and inputted to Weka in an .arff file. Weka then uses J-48 classifier to classify the products by level.

Successes:

-Data input

-Classifier runs well on simulated data

Necessary Improvements:

-Classifier must always classify the product number to the level based on the first 2 attributes

-Classifier must use only historical data as training data, and then classify the remaining, that is, only the data of the 6 products for this year. This is tricky, especially making it easy.

**Design and Technical Approach**

I would like to use MATLAB or Python. Data will be collected from Euromonitor and other business databases available through the Kogod School of Business here at American University. Data will be downloaded to excel, reformatted and inputted into Python or MATLAB.

Either I will write my own classification algorithm or I will use WeKa. I don’t know which features are best so I would like to discuss this with Dr. McCoy. From reading I’ve learned that to provide the best learning, the algorithm should consider representation, evaluation, and optimization of the problem.

**Documentation and Access**

This is an individual project so code and documentation will be stored on Brian’s personal laptop.

**Evaluation**

This depends on the representation scheme selected. It can consist of:

Accurary/Error Rate, Precision and recall, Squared error, Likelihood, Posterior probability, Information gain, K-L convergence, Cost/Utility, Margin

**Plan for Deliverables**

*Checkpoint Report*

Training data set from business databases

Test data set from business databases

Machine Learning code from Python or MATLAB or WeKa

*Final Report and Project*

Document describing the algorithm used and the representation, evaluation, and optimization scheme used along with all finalized documents from Checkpoint Report.

**Separation of Tasks for Team**

Defining relevant variables: Brian

Data collection: Brian

Data formatting: Brian  
Machine learning code: Brian

Evaluation code?: Brian

Inline documentation: Brian

Final Project Summary: Brian