Han Jun Won
100848402
COMP 4106

Wine Testing by Intelligent Machine

Problem Domain

Wine is one of the most widely drunk alcoholic beverages in the world. It has grown into a multibillion dollar industry and is consumed worldwide in almost every country. Because of the large growth in the industry competition is fierce. For wine companies not only are the sale numbers important so is the quality of the wine and the public/critic's opinion of it. These companies have decades of experience but they continue to experiment and tweak their recipe in the hopes that they can produce higher quality wine than their competitors. The industry standard for wine testing is internal testing, limited critic test and then the public release test. In the internal test, people directly involved in the company taste and rate the wines that they produce and see if it is up to par. The next phase, limited critic test, is when the wine is sent to several critics and they write reviews and score the wine. The last phase is when the wine is sold to the public and everyone can get their hands on it. Any critic or consumer can then taste the wine and review it.

Problem

Most companies believe that the second phase is the most important in the testing phase. Good early reviews can lead to large sales and a better reputation for the brand, while bad reviews will hinder sales greatly and hurt brand image. So the problem that companies face is how do you predict or replicate the results of the second phase during the first phase. If they are able to do so then they could stop batches that are harmful to their reputation before they reach the hands of critics.

Motivation

My motivation for this problem is my fascination of the application of artificial intelligence to subjective experiences. Basically, how well can AI predict subjective scores relying solely on objective data. I thought wine would be a perfect example for such application as wine has physicochemical qualities that are purely objective but its score is based on taste and scent which are purely subjective. Another motivation I had for this problem is that although I am interested in wine, I am not that knowledgeable of it. I do not know what kinds of wine I like nor can I accurately describe the taste of the ones I taste, all I can do is say how much I like it. So I thought that by solving the above problem I could create my own dataset of my wine preferences and be able to make knowledgeable decisions on what wine I should buy.

Techniques

The two techniques I hoped to use to solve this problem were Artificial Neural Networks and Genetic Algorithms. Since this problem is largely a data classification problem I thought that these two would be the most fitting.

Artificial Neural Networks (ANNs) are a family of learning algorithms that are inspired by biological neural networks. ANNs have been used in machine learning to teach various things. They have tremendous uses in Pattern Matching, Data Clustering and Optimization and have been known to produce fairly accurate results in all fields. I would be creating an ANN that took in the physicochemical properties of wine that output a score based off those properties.

I was advised to try genetic algorithms (GAs) as well during the discussion of my project but in the end I decided that it would not help for data classification as GAs are used more for optimization and a good fitness function for a chromosome based off the physicochemical properties cannot be found.

Design Choices

Some of the design choices for my ANN were very straightforward while some were difficult to decide upon. From the very start I knew that I required a Feed Forward Neural Network. I was inputting several different types of data and expecting an output based solely off those data. I was not intending to use the output in any other meaningful way other than to check accuracy so an FFNN made the most sense.

Next I decided upon the use of a multilayer perceptron model instead of a single layer perceptron model. This means that there is at least one layer of neurons between the input neurons and output neurons. The multilayer perceptron model was preferable compared to the single layer perceptron model as single layer perceptron models can only learn linearly separable patterns and due to the complexity of wine I thought it would be more likely that it would be nonlinearly separable than not.

Since it was a FFNN with a multilayer perceptron model I knew that it was gonna be relying on supervised learning. This means that I had to provide the network with a large enough dataset so that it could “learn” ie. adjust the weights in the network accordingly to fit the given dataset. Then it could try applying the network to a test set to see how accurately it could classify the data. The accuracy of testing on the test set would hopefully replicate its real world use in classifying wines based on their properties. The test set I used is from the Center for Machine Learning and Intelligent Systems and is based off of white wine from Vinho Verde. The data set was created by P. Cortex and others. Check reference section.

I had difficulty choosing between a top down or a bottom up approach for the design of the network. A top down approach starts off by building a large network and then breaks it down one by one until there is a hit to performance/classification accuracy. A bottom up approach is the exact opposite of the above. It begins with a very simple network and adds neurons one by one until the performance plateaus. In the end I went with the bottom up approach as I was inexperienced in neural networks and could not determine what a large enough network size would be to begin the top down approach.

I decided on the most commonly used technique for learning in a multilayer perceptron model, backpropagation. After the input data propagates through the network into the output neuron the backpropagation algorithm first checks the error margin between the output and the target score. It then propagates that error difference backwards through the network and adjusts the weights according to the gradient descent function.

Application

Most of the design choices mentioned above worked well together. Through the bottom up approach I managed to create a FFNN that had two hidden layers between the input and output neurons. The first consisted of eight neurons and the second with four. After this it seemed like the classification accuracy stayed consistent for the test set.

The test set was can be manually selected. The 4800 entry dataset was divided into 10 clusters 480 entries. At any time you can select which cluster will be used as the test set. The remaining entries are then used to train the network.

The ANN itself seems to produce rather accurate results. After 40 ticks it provides an average classification accuracy of about 80%. After 60 ticks the test accuracy for some clusters decrease which means that the neural network has been trained to overfit the learning set.

Classification Accuracy (ticks vs cluster)										
	1	2	3	4	5	6	7	8	9	10
1	43	39	41	48	38	40	45	44	50	81
20	68	71	56	75	73	74	69	71	73	82
40	81	73	73	80	82	82	79	80	83	83
60	82	74	74	79	82	81	80	81	84	84

Possible Enhancements

The biggest enhancement this project can get is to be able to receive dynamic data information through a chemical analyzer of some sort. This way after learning on the data set the network can be directly applied to different wine and test them on the spot.

Another enhancement that could be done stems from my second motivation for the problem. Being able to insert a custom data set based on own preferences would allow the neural network to train on that and provide score for new wine tailored to one’s taste.

Lastly, the input data can be widened considerably. Here the eleven inputs were all physicochemical properties but wine is dependent on more than that. If data such as grape type, year of harvest, harvest month, price and such were provided I believe the neural network would provide much more accurate results. The dataset says that these types of data were unavailable due to privacy and logistic issues.

Reference

P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009

Riul, Antonio, et al. "Wine classification by taste sensors made from ultra-thin films and using neural networks." *Sensors and Actuators B: Chemical* 98.1 (2004): 77-82.

Sun, Li-Xian, Klaus Danzer, and Gabriela Thiel. "Classification of wine samples by means of artificial neural networks and discrimination analytical methods."*Fresenius' journal of analytical chemistry* 359.2 (1997): 143-149.

Appendix

Program was built using NetLogo 5.2. This version is required to run the program.

1. Run Program
2. Select test cluster
3. Select learning rate
4. Press Setup
5. Press Train