GAINS: Genetic Algorithms for Increasing Net Sales of a Mobile Reverse Demand Communication System

Michael H Wolk
Department of Computer Science
Colby College
Waterville, ME 04901
mhwolk@colby.edu

ABSTRACT

In this paper, I describe MRDCOM, a mobile reverse-demand communication system for the pizza industry. The web-based system will support student (buyer) sign-up and will be capable of sending buyers discount food offers through mobile phone text messaging technology. Pizza sellers will be able to easily purchase pizza sales offers that will then be sent only to a relevant sub-group of users. A component of MRDCOM, GAINS, is an offline genetic algorithm (GA) learning component designed to model the offers that are accepted. The rule set derived from the GA component will be used to raise user satisfaction by sending offers to users most likely to accept them. GAINS is expected to increase the accepted offer rate will provide economic benefits for students, pizza sellers, and the administrators of the MRDCOM system.

Categories and Subject Descriptors

I.2.6[**Artificial Intelligence**]: Learning.

General Terms

Algorithms, Design, Economics, Experimentation, Human Factors.

Keywords

food, Genesis, genetic algorithms, Internet, pizza, Ruby On Rails.

1. INTRODUCTION

1.1 Motivation

College students consume a lot of pizza. Students are constantly ordering pizza products for delivery, and most colleges have a considerable number of pizza sellers clustered within the general vicinity of campus. Like any industry, pizza restaurants are sometimes subjected to drops in demand that can lead to lower than expected revenues. The proposed reverse demand communication system is designed to help mitigate revenue losses by facilitating the communication of discount product offers to students' mobile phones.

GECCO'07, July 7-11, 2007, London, England, United Kingdom.

Some of the system design choices have been guided by the goal of offering a product that is economically advantageous to student buyers, pizza sellers, and the operator of the system.

Therefore, the intent is to offer a strong return on investment (ROI) to pizza restaurants and to provide student buyers with a good deal. It is in the best interest of pizza restaurants and the system operator to retain as many users as possible.

The more users, the greater the likelihood of finding individuals interested in buying pizza at all times. I propose a business model of charging a small fee per number of pizzas that a restaurant chooses to sell. Thus, it is advantageous to investigate how to minimize student buyer account cancellations (rate of attrition).

In particular, I wish to investigate ways to send pizza offers primarily to students likely to find them relevant at a particular point in time. Most mobile phone owners must pay to receive text messages in the United States, and I want to diminish attrition by sending offers they are likely to consider relevant.

1.2 Project Goals

The Mobile Reverse Demand Communication System (MRDCOM) is a testable prototype capable of facilitating business-side offer purchases and buyer-side responses to offers. I have added Genetic Algorithms for Increasing Net Sales (GAINS) to MRDCOM, to investigate the extent to which genetic algorithms might be used to assist in determining who and how many people should be sent a given offer.

Figure 1. A cheese pizza like those offered through MRDCOM.

2. BACKGROUND

I have not been able to find a similar system to the reverse demand communication system, but there are web-based systems with related features. AccuWeather, a weather forecasting company sells "AccuWeather.com Messenger™", a pay text message service that sends weather forecasts to subscribers' mobile phones. While subscription statistics are unavailable, this suggests there is a market for services utilizing text-messaging technology.

Businesses such as Priceline.com and Hotwire.com are examples of alternative sales systems that have succeeded in the marketplace. Priceline's 2006 gross profit of over $400 million USD suggests a significant level of consumer adoption. The mobile reverse demand communication system is not a so-called opaque sales system like Priceline's Name Your Own Price® product, but the hotel rooms sold on Priceline are discounted because of low demand much like the pizzas offered through MRDCOM. In the case of low demand for pizza, however, sales must be made within minutes instead of the lag of several days possible for hotel room sales.

Figure 2. Priceline.com offers steep discounts through a non-traditional purchasing system, 2007.

The use of GAs was supported by literature [1] which suggested their viability for classification tasks. I also found further materials [2] which supported this assertion.

3. SYSTEM DESIGN

3.1 Ruby on Rails

The Ruby on Rails framework [3] is a web development framework written in the object oriented Ruby Programming Language. Ruby itself is simple to use, and I found Ruby code easier to modify than alternatives like Java because of the fewer characters needed to express concepts as well as the code's higher readability. The Ruby on Rails framework was chosen because of its many built-in features. While there is a learning curve, I believe that I have saved time by using the framework to implement the mailing system and some of the basic create, read, update, and delete (CRUD) functionality of the system. Furthermore, I have taken advantage of the framework's ActiveRecord system. By using ActiveRecord, I was able to set up a back-end SQL database without having to write one line of SQL. The time saved there has been invested in implementing more sophisticated parts of the system.

3.2 Genesis

I am using Genesis [4], the Genetic Search Implementation System, to implement the GAINS component of MRDCOM.

3.3 System Implementation

The system was designed to be visually appealing to users familiar with Web 2.0 sites. I modified an external layout in order to achieve my aesthetic goals. Overall, I designed simple site pages and maintained a minimalist design philosophy. The system is not intended to be a content provider or a destination site. Its dual purposes are to allow users to enter and update their personal preferences and to provide a place for students to quickly accept offers.

3.4 Data Models

Colleges are defined by a college name, zip code, city name, and state abbreviation. They are related to dorms, which belong to a specific college. A dorm is defined by a name and the key value of its associated college. I implemented the system so that multiple colleges can have dorms with the same name, but no college can have more than one dorm with a specified name. While the use of zip codes and state abbreviations limits the system to the U.S. university system, it would be trivial to modify the college model to accommodate for schools in places outside of the United States like Canada. This change, however, is beyond the goals of the current system.

Mobile providers must be modeled so that I can build the email addresses needed to send text messages. A text message can be sent to a mobile phone by sending an email to the concatenation of the phone number and a carrier-specific string. As a result, buyers must select a mobile phone provider during the sign-up process. The mobile provider data model contains a name for the phone provider and an email string which is the text appended to a user's phone number in order to produce a valid email address.

A student model representing a student buyer holds relevant data including college, dorm, mobile phone number, mobile phone provider, and preference information on when the user is willing to accept offers. The student model also includes demographic information including computer operating system, class year, gender, and academic area of study. Academic area of study is more broad than a major. They include: The Arts, Natural Science, Social Science, Mathematics, and Engineering.

For each offer, a user who is sent a text message is given an offer transaction entry. These entries specify the seller of the pizza, the time of the offer, and whether or not the user accepted the offer.

3.5 GAINS and Algorithms

The prototype system has been simplified to sell only cheese pizzas at one price. This simplifying assumption made it possible to get a basic version of MRDCOM running within a shorter time-frame than if I tried to implement everything at once. Later versions, however, will possibly include variable pricing and a range of products.

A simple pizza selection algorithm would send an offer to a portion of the users who have indicated in their time preferences that they are interested in receiving offers.

Simple Algorithm Use Case

1. A pizza restaurant requests to sell n pizzas to college x within the next ten minutes.
2. The system searches the user database and selects all students of college x who have agreed to receive offers at the current time.
3. The system selects a subset of the students from Step 2 to receive offers.
4. Offer text messages are forwarded to the students selected in Step 3.

Step 3 of the simple algorithm is intentionally vague. One basic implementation for the selection process would be to randomly choose a subset of students to receive an offer. A random algorithm might at times not send enough offers that are needed to sell all pizzas while at other times it might send too many. One significant issue is that this algorithm might consistently send offers to a user who always declines them. A more informed implementation could use user data and past behavior to better predict the number of offers to be sent as well as the users who should receive them. GAINS uses genetic algorithms to analyze past behavior and help predict future user decisions.

MRDCOM is configured to generate a file containing data corresponding to each offer message sent to student users. The data captures time information, a user's demographic data, and whether the offer was accepted or ignored.

Table 1 shows the structure of the bit strings that GAINS uses to model the offer data. The day of the week and time are encoded as simple binary strings. Days are encoded using a string of seven bits where one is selected. “1000000” represents Monday and “0010000” refers to Wednesday. The hour of the day is noted the same way but with a 24 bit string. This bit encoding allows the GA string to specify, for example, a subset of offers on Mondays and Wednesdays at 7, 8, and 9pm, specifying more than one possibility for each attribute. A student's class year has six possible options: Freshman, Sophomore, Junior, Senior, Graduate, and Other. A PC Preference attribute captures whether a user claims to use a Windows, Mac, or Other operating system. Students are required to select their major area from a list of general areas of study. Location types describe the surrounding environment of a school and include: rural, town, city, and big city. Finally, the bit string states whether the offer was accepted. The GA string totals 52 bits.

Table 1. The GA string representation

0	1	2	3	4
Day Of Week (7)	Time (24)	Class Year(6)	Gender(2)	PC-Pref(3)

5	6	7
Area of Study (5)	Location-Type(4)	Offer Accepted(1)

The use of class year, gender, and especially computer preference may seem strange. Including class year makes it possible to account for different user behavior over time. If, for example, younger students are more comfortable with using MRDCOM, this is a way to capture this difference in behavior. Similarly, it is possible that there is some difference in buying habits between genders and accounting for this information may help improve the user selection process. Finally, PC preference is a way to indirectly divide users into different groups. There has been research [5] supporting differences in the buying behavior of PC and Mac users. It might be the case that there are unique buying and lifestyle characteristics for owners of different types of computers. Perhaps Linux users stay up all night and love pizza. If this is the case, the PC preference variable will help to capture these behaviors.

The GAINS component of the system consists of learning a rule set to describe the buyer groups most likely to accept offers under different circumstances. GAINS uses the GA string representation illustrated in Table 1 to evolve descriptions of subsets of the data; odds ratio [6] is used as the fitness function. As a preliminary step in development, GAINS finds the one best rule (describing the strongest subset in the data); future work will extend GAINS to find complete rule sets. The data for GAINS consists of information about a specific offer and the user who was sent the offer. This includes the time of the offer, whether the offer was accepted, the school where the offer was sent, and demographics on the user. These consist of the demographic qualities of the student model.

4. RESULTS

Although still in development, GAINS is expected to increase the relevance of system offers by sending the right offers to the right student users. At this time, I do not have real use data to test GAINS and instead am testing if it is capable of making these distinctions using simulated data. I am creating a large artificial data set of user behavior and plan on giving different demographic groups arbitrary behavioral patterns. For example, city dwelling male Mac users might generally accept offers from only nine to eleven in the evening. By creating different demographic profiles, I wish to test and refine GAINS to determine if it is able to identify patterns that can improve the offer group selection process.

5. CONCLUSION

MRDCOM is a unique solution to a problem that exists in many industries. My implementation is designed to target the college pizza market. Yet, there are other markets, such as the electrical power industry, that could also benefit from software that is capable of linking price-sensitive buyers and sellers during momentary drops in demand.

I believe that a large student user base is key to making MRDCOM a success. Furthermore, I would like to have a diverse population of users so that pizza restaurants have a better likelihood of selling pizzas at all times. These two qualities are the motivating forces behind GAINS. With the GAINS component, the system can learn how different demographic characteristics affect behavior and then use this knowledge to guide the selection of users to receive offers. If GAINS can learn useful rule sets, it may be able to mitigate attrition and therefore improve MRDCOM.

6. ACKNOWLEDGMENTS
I would like to thank Clare Bates Congdon and Adam Fischbach for all their help and support. I would also like to thank the developers of Ruby, the Ruby on Rails framework, and Genesis. The cheese pizza photo is by flickr user Roboppy and is licensed under a Creative Commons Attribution-NonCommercial 2.0 license.

7. REFERENCES

[1]Goldberg, D. E. 1989 Genetic Algorithms in Search, Optimization and Machine Learning. 1st. Addison-Wesley Longman Publishing Co., Inc.

[2] Holland, John. Genetic Algorithms, in Scientific American, July 1992.

[3] Ruby on Rails web site. web-based information repository, 2007. http://www.rubyonrails.org

[4]J. J. Grefenstette. A user's guide to GENESIS. Technical report, Navy Center for Applied Research in AI, Washington, DC, 1987. Source code available at http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/genetic/ga/systems/genesis/.

[5] Fried, Ian. Are Mac users smarter?, News article, CNET News.com, 2002. Article available at http://news.com.com/2100-1040-943519.html/.

[6]M. S. Kramer, Clinical Epidemiology and Biostatistics: A Primer for Clinical Investigators and Decision-Makers. Berlin: Springer-Verlag, 1988.