A Multiagent System Approach to Grocery Shopping

Hongying Du and Michael N. Huhns

Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, 29208 USA

Abstract We present an approach to social grocery shopping based on customers trading information about item prices and quantities in order for the customers to find the lowest prices for the goods they purchase and the most convenient plan for buying them. Because collecting and reporting prices is tedious, agents representing customers are needed to make this approach practical. Agents also have the potential to learn which other agents can be trusted. We use a realistic shopping list based on the U.S. Consumer Price Index in order to guarantee the realism of our results. By visiting actual grocery markets and comparing prices, we have discovered that the total cost of a list of groceries can vary by 13%. We have also discovered that by shopping optimally, that is, buying each item from the cheapest store, the result can be a savings of 16% over shopping at the store with the lowest total cost. To shop at minimum cost requires customers' agents to report prices to each other. If they do, each customer is likely to achieve at least a 10% savings. However, what if the reported prices are inaccurate? Would customers be worse off than if they just shopped randomly? We investigated the robustness of our multiagent shopping system in the presence of errors in reported prices. From this, we determine the potential savings an average customer might obtain.

1 Introduction

Aided by information systems for analyzing customer buying data, supermarket chains continually alter the prices of items to maximize their profits. They do this by, in essence, experimenting on their customers. For example, the price of an item might be raised at one store until customers stop buying it. This maximum price is then used at all of the stores in the chain. Customers however, do not have any comparable information systems that might aid them in price comparisons and are often at the mercy of the stores. Most stores do not post their prices online, so that customers have to visit each store to find the prices of groceries, which makes comparison shopping prohibitive.

Imagine an online system where customers could post the prices paid for groceries and where a prospective shopper could enter a grocery list and obtain a pointer to the store with the lowest total price. This would enable comparison shopping for groceries and would render the customer-to-store interactions fairer. It would also encourage stores to offer their true prices to avoid driving away potential customers. However, the effort required from the customers would be substantial. To make the effort reasonable and manageable, each customer could benefit from an agent that represented his/her interests and interacted with the agents of the other customers and, possibly, with store agents.

However, there is an expense in implementing and operating such a system. Moreover, its success is dependent on prices entered by other customers, on the availability of goods, and on prices that stores might change to yield an advantage for them to the disadvantage of customers. Hence, it is subject to errors and manipulation. To be feasible, the potential cost savings must substantially exceed the expense and effort of its implementation.

In this paper, we investigate the efficacy of a customer-oriented comparison-shopping system for groceries and the trade-offs in an implementation of it. Our approach is to use real data, normalize it according to typical customer actions, and simulate a system of stores and customers. We introduce both random and systematic (manipulation) errors into our simulation in order to evaluate its robustness. We provide a customer with the best combination of price and quality for a list of products available at different stores and recommend which store or stores would be optimal for shopping.

2 Background

Price comparison services (also known as comparison shopping services) allow people to query a product's prices at online stores. The services list the product's prices in all of the stores and sort the prices to provide customers with support for their online shopping. An intelligent software agent to implement comparison shopping is called a *shopbot* [1]. The first well-known shopbot, BargainFinder [2], provided comparison shopping for music CDs. It searched eight online music stores and displayed all prices on a webpage. Customers gained obvious benefit from BargainFinder and it has been used widely. Current shopbots have greater functionality than before by including information about shipping expenses, taxes, vendors' rates, and product reviews. The app RedLaser [3] accepts the barcode of a product from an iPhone's camera, searches many online stores, and shows their prices on the phone.

There are typically three steps for a shopbot to deal with data. First, it retrieves data from online stores or other shopbots, possibly by using an extraction method as in [4]. Second, the data is processed according to a user's command. Last, the results are shown to the user on a webpage. Other researchers are developing algorithms to improve the behavior of shopbots [5, 6] and make them more robust to changes in the stores' websites, such as by using Semantic Web concepts [7].

3 Analysis

There are a number of variables in grocery shopping. Our simulation of it uses five parameters: customer input, customer location, store location, item price, and item quantity. Customer input is a shopping list of the items a customer wants to buy and their quantities. Store location and customer location are used to calculate the fuel cost when driving to and from the stores. Item prices are those either reported by customers or by stores. We assume the quantity of a specific item in a store is either zero or infinity.

Our algorithm begins with a customer's shopping list of items and quantities. If the customer just goes to the stores with the lowest price for each item, the customer might need to go to many stores and spend more on fuel. So we search in all the stores and find the lowest and second lowest prices of each item on the list. If the prices of an item in several stores are the same, we consider the item with best quality first. We considered all possible combinations of the two prices and calculate the total cost as the sum of grocery cost and fuel cost. When calculating the fuel cost, we assume the customer's path is to go to the nearest store he needs to go to where he has not already shopped until he gets all the items. For comparison, we also calculate the cost if the customer chooses to go to stores using three other strategies: (1) choose one store randomly and buy all the items at that store, (2) go to the nearest store, or (3) randomly go to one of the five nearest stores. Then we calculate the ratio of the total grocery and fuel cost of these three methods over that of our optimal multiagent method.

Last, we evaluate robustness. There are two ways price information might be erroneous. First, if we rely on the stores themselves to report the prices, they might claim their prices are lower than they actually charge. Second, customers' agents can report prices by querying the RFID tags of the items the customers bought, but the agents might make mistakes when they acquire/report prices. We investigated both in our simulations.

4 Grocery Shopping Simulation

We used the Netlogo platform [12] for our grocery shopping simulation, which we separate into two phases. In the first phase, we simulate shopping according to fictitious prices and qualities generated randomly and examine the ratio of the cost of other methods over that of our method and evaluate the influence of different values for the parameters. For each combination of parameter values, we ran the simulation 100 times and used the mean of the 100 results. To consider deception, we assume that the deceptive stores say their prices are 10% lower than the real prices and the percentage of deceptive stores are 25%, 50% and 75% separately to see how the results change.

In the second phase, we use real prices collected from South Carolina stores to see whether there is a significant difference between using fictitious prices and real prices in the simulations. With real prices, the store location, item price, and item number are fixed. We assume here that an item's qualities are the same in all stores. However, quality information can be used in practice by allowing customers to rate it. For the customer input, we constructed a shopping list according to the U.S. Consumer Price Index (CPI). The CPI measures a price change for a constant market basket of goods and services from one period to the next within the same area (city, region, or nation) [13]. Along with the CPI, the relative importance of the items in the market basket is published. We created a realistic shopping list by selecting an item from each category according to its relative importance [14]. Since there are many categories, we selected a representative list of 33 items. For these, we collected item prices from 5 different stores. Table 1 shows a few of them. The complete price list can be found at [15]. We compared the savings when a customer went to two stores to buy goods than when the customer went to just one store. To measure robustness, we checked the results when there was a 10% possibility that the customers reported each digit of the prices wrong. When a desired item isn't available in a store, such as apples not available in store 4, we assume that the customer will go to another store to buy it.

Table 1. The shopping list

Item	Walmart	Publix	Food Lion	BI-LO	Target
	store 0	store 1	store 2	store 3	store 4
Tropicana: orange juice, 64oz	2.92	3.79	2.97	3.69	2.99
Simply Orange juice, 1.75l	3	3.79	2.99	3	2.99
Corona extra: 12oz*6	8.47	8.29	7.99	8.29	6.5
Budlight: 12oz*6	6.97	6.49	5.99	6.99	5.25
Totino's: pepperoni pizza, 10.2oz	1.25	1.49	1.67	1.67	1.2

5 Results and Discussion

In our simulation, we assumed there were 12 stores and 30 kinds of items. Given 10 items a customer wants to buy, we ran the simulation 100 times for a random change in one of the simulation parameters and calculated the mean, as shown below. We assumed the parameters are independent, so our simulation varied them one-at-a-time. The values in the table are the ratios of the total grocery and fuel cost using an alternative method over our optimal multiagent method.

The simulation results show that our approach to deciding at which stores to shop can save 21% or more, except when customers change their shopping lists. Our approach is better than the other 3 methods for all cases. When a shopping list changes, the savings are lower, possibly because the randomly generated customer input may contain fewer items. The consideration of fuel decreased the savings by only 2.4%, so we did not consider fuel cost for our results using real price data.

Table 2. Simulation results using randomly generated price data

	Mean Ratio of Shopping Method to Optimal Multiagent Method		
Simulation Parameters	Choose Store Randomly	Choose Nearest Store	Choose 1 Store Randomly from 5 Nearest
Vary Customer Location	1.2328	1.2365	1.2178
Vary Store Location	1.2351	1.2325	1.2269
Vary Item Price	1.2150	1.2180	1.2225
Vary Number of Items	1.2637	1.3317	1.2911
Vary Shopping List	1.1732	1.1080	1.1573

What if 25%, 50%, 75% stores are deceptive by claiming that their price is 10% lower than the real price, thereby luring customers to shop at the wrong stores? Our simulation chose deceptive stores randomly. The results were that a customer would save 5.2% less when 25% of the stores are deceptive and 9.1% less with 75% deceptive stores.

Using the real price data we collected (Table 1), the total cost of the goods on the shopping list if a customer shops at just one store varied from 114.27 from store 0 to 129.52 at store 1.

The cost of buying each item at its lowest price is 98.44, which is more than 13% lower than going to one store, but a customer would have to go to four stores to get this lowest price. Because a customer might not want to go to more than two stores, we tried all combinations of two stores and calculated the cost. The lowest cost of 106.58 is 6.7% lower than going to just one store.

What if the customers reported the price data wrong? We simulated this situation by giving each digit of a price a 9% possibility to change to another digit, each with a 1% possibility. When the price information is wrong, the only thing changed are the stores the customer would go to, because the customer would still pay the real price at the store. We ran the simulation 500 times and found there is a 2.2% possibility that the customer would go to another store due to wrong prices, rather than going to the store with the lowest price. The average cost is 114.57, which is very close to the optimum 114.27. For the results with two stores, there is a 37% possibility that a customer would go to stores other than the best combination of two stores. Though the possibility is significant, the average cost is 107.04, which is very close to 106.58, the lowest price possible for two stores. So on average, a customer can still save 6.3% by going to two stores compared to just one store, even if the prices are incorrect.

6 Conclusion

A societal grocery shopping system as described in this paper would be useful and practical, because it helps customers obtain a savings of 21% or more according to our simulation. Even with deceptive pricing by stores or incorrect price data reported by other customers, it will still produce savings. We considered five parameters that characterize real shopping experiences: customer location, store location, item price, item number, and customer input. We varied them in our simulation to explore this five-dimensional space and produced results consisting of the average savings achieved by customers. To validate our results further, we also used real price data in a simplified version of our simulation containing fewer stores and shopping at just two of them. The results indicate an average savings of 6.7% by choosing the best two stores. Even with incorrect price data, customers can still save 6.3% on average. An implementation of our approach would require a multiagent social infrastructure where agents representing customers could report prices they discovered and use prices reported by others. Based on both simulated and real data experiments, and the expected costs of such an infrastructure, our system would be useful and cost-effective in practice.

References

1. Clark D (2000) Shopbots become agents for business change. IEEE Computer 33(2): 18 - 21. doi: 10.1109/MC.2000.820034
2. Krulwich B (1996) The BargainFinder agent: Comparison price shopping on the Internet. In Williams J, editor, Agents, Bots and Other Internet Beasties, SAMS.NET.
3. http://www.redlaser.com/.
4. Yang J, Kim TH, Choi J (2005) An Interface Agent for Wrapper-Based Information Extraction. 7th Pacific Rim International Workshop on Multi-Agents (PRIMA' 04): 291-302, Springer. doi: 10.1007/978-3-540-32128-6_22
5. Tang Z, Smith MD, Montgomery A (2010) The impact of shopbot use on prices and price dispersion: evidence from online book retailing. International Journal of Industrial Organization 28 (6): 579-590 doi: 10.1016/j.ijindorg.2010.03.014.
6. Greenwald AR, Kephart JO (2000) Shopbots and Pricebots. Agent Mediated Electronic Commerce II (AMEC' 09): 1- 23, Springer. doi: 10.1007/10720026_1
7. Lee HK, Yu YH, Ghose S, Jo GS (2004) Comparison shopping systems based on semantic web – a case study of purchasing cameras. Grid and Cooperative Computing (GCC' 03): 139-146, Springer. doi: 10.1007/978-3-540-24679-4_27
12. Kawa A (2009) Simulation of dynamic supply chain configuration based on software agents and graph theory. International Work Conference on Artificial Neural Networks (IWANN' 09): 346-349, Springer. doi: 10.1007/978-3-642-02481-8_49
13. BLS Information, Glossary, U.S. Bureau of Labor Statistics Division of Information Services. http://www.bls.gov/bls/glossary.htm#C. Accessed 4 October 2010.
14. (2007-2008 Weights) Relative importance of components in the Consumer Price Indexes: U.S. city average, U.S. Bureau of Labor Statistics Division of Information Services. http://www.bls.gov/cpi/#tables. Accessed 4 October 2010.
15. http://www.cse.sc.edu/~huhns/ShoppingList.html