

Decision-theoretic Learning of Agent Models in an Information Economy

Christopher H. Brooks and Edmund H. Durfee *

Artificial Intelligence Laboratory
University of Michigan
Ann Arbor, MI 48109
{chbrooks,durfee}@umich.edu

Abstract

We demonstrate how a producer of information goods can use a successively complex series of models to learn the preferences of consumers efficiently. We provide metrics for estimating the precision, accuracy, and learning complexity of different models, thereby providing a producer with the metrics needed to apply decision theory in selecting a sequence of models. We present experimental results demonstrating the effectiveness of this approach, and discuss current research on extending this idea to learning preferences over categories or strategies of other agents.

Introduction

The combination of electronic distribution of information goods via the World Wide Web and the use of automated agents to buy and sell goods on behalf of human users has led to the study of agent-based information economies. Our previous work in this area has examined the problem of how sellers in an information economy can learn the preferences of a consumer population (Brooks *et al.* 1999; Brooks, Durfee, & Das 2000). In particular, we study problems where the consumer population is nonstationary. We assume that each information goods producer is interested in maximizing its aggregate profit. Since a producer will only have a limited number of iterations to interact with a consumer population, it is faced with a dilemma: what should it learn, and how long should it learn, so as to maximize aggregate profit?

In this paper, we present a decision-theoretic approach that an information goods producer can use to efficiently learn consumer preferences. It is based on the idea of *model refinement*. Roughly speaking, a producer sequentially constructs a set of models of varying accuracy and complexity. It begins with a simple model and then switches to more complex and accurate models as information is acquired. The difficult aspect of this problem is determining how to compare models. We introduce three metrics, accuracy, precision and complexity, for estimating how many data points are needed to learn a particular model to a given degree

of accuracy. Additionally, since a producer is making a sequence of modeling decisions, it must be able to estimate how data acquired in learning one model can be applied to other models. We also describe heuristics that an information goods producer can use to make this estimation. We show the decision procedure that a producer should go through to select a series of models, and present empirical evidence for this strategy's effectiveness.

Modeling

We assume that a producer has access to a set of N articles that it can offer for sale to a consumer population. Consumers may choose the articles they want from this set, subject to the price schedule the producer sets. Producers then have the problem of selecting a price schedule and prices within that schedule.

In this work, we have considered six price schedules: linear pricing, where consumers pay a fixed price for each article, pure bundling, where consumers pay a fixed price for all N articles, two-part tariff, where consumers pay a subscription fee plus a per-article price, mixed bundling, where consumers have a choice between a per-article price and a bundle price, block pricing, where consumers pay a per-article price of p_1 for the first j articles and per-article price p_2 for the remaining articles, and nonlinear pricing, where consumers pay a different price for each article.

We model consumers using a model originally introduced by Chuang and Sirbu (1997). The model consists of two parameters: w , which indicates a consumer's value for its most-preferred article, and k , which indicates the fraction of the N articles available for which the consumer has a positive valuation. The valuation $V_j(i)$ of the i th most-preferred article by consumer j is a linear function of these variables, expressed by:

$$V(j) = \begin{cases} w_j(1 - \frac{i-1}{k_jN}) & \text{if } i-1 \leq k_jN \\ 0 & \text{if } i-1 > k_jN \end{cases} \quad (1)$$

Even though a single consumer's valuation function is quite simple, if consumers are heterogeneous in either w or k , the aggregate demand will contain nonlinearities. If we plot the number of articles purchased against consumer valuation of that quantity of articles, we can visualize price schedules as fitting a nonlinear curve, with profit being the area under that curve. Figure

*This work was supported in part by an IBM University Partnership Grant and by the National Science Foundation under grant IIS-9872057.

Copyright © 2001, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

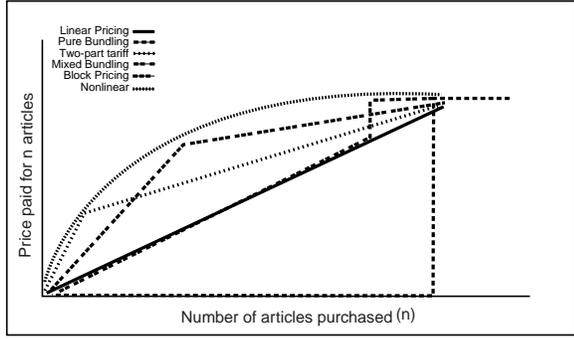


Figure 1: An illustration of price schedules as approximations of a nonlinear curve.

1 gives an illustration of this, with consumer demand denoted as 'Nonlinear' (the nonlinear pricing schedule can fit demand exactly), and other schedules capturing different portions of the available profit.

Model Refinement

We refer to the sequential selection of progressively complex models and accurate models as *model refinement*. To describe model refinement, we must introduce some terminology.

We begin with the consumer population's *reaction function*. This is a function $\pi : \mathbf{p} \rightarrow \mathbb{R}$. \mathbf{p} is a price vector, indicating the amount charged for each article offered. This function indicates the amount the consumer population is willing to pay for any offering. We refer to a particular \mathbf{p} as an *input set*. Typically, a producer will only be concerned with modeling a reaction function over some particular input sets.

A *model* is a representation of this reaction function. Each model has an input set i that maps to an output in \mathbb{R} . All models which use the same inputs are referred to as a *model family*. For example, there is the family of linear pricing models, the family of pure bundling models, and the family of nonlinear pricing models. Within a family, models differ only in the particular values assigned to their parameters.

Model families have two characteristics of interest. The first is *precision*. Precision indicates how closely the best model in a family approximates the reaction function over a set of relevant input sets. In order to measure precision, we must know both how different a model is from the reaction function for each relevant input set, and also how likely those input sets are to occur. Let S be a collection of input sets, and $Pr(s)$ be the probability of a particular input set $s \in S$ occurring. We can then compute the precision p_{MF} of a model family MF as:

$$p_{MF} = \underset{M \in MF}{\operatorname{argmax}} 1 - \sum_{s \in S} \frac{|M(s) - \pi(s)|}{\pi(s)} Pr(s) \quad (2)$$

In other words, this is the fraction of available profit that the best model in this family can hope to capture over the relevant input sets. (If S is continuous, we integrate rather than sum.)

We would also like to be able to measure how good a particular model is, relative to the optimal model of its

family. We refer to this as a model's *accuracy*. For a given family of models MF , let M' be the model within this family that maximizes precision over a collection of input sets S . We can then define the accuracy a_M of a given model M as:

$$a_M = 1 - \sum_{s \in S} \frac{|M'(s) - M(s)|}{|M'(s)|} Pr(s) \quad (3)$$

The third factor we must consider in comparing model families is the expected time (measured in the number of samples) needed to learn a model from that family. We refer to this as a model family's *complexity*. The complexity C of a model family estimates how good a solution is (as a fraction of optimal), as a function of the number of data points observed. Formally, $C(MF, n, S) = a_M$ says that, after n price/profit points are observed using a model $M \in MF$ with input sets from S , the accuracy of M is a_M .

Empirically, we have found that a sigmoid function works well for approximating complexity, at least when applied to pricing data. The profit landscapes tend to be composed of large plateaus, dotted with hills that are easy to locate and ascend initially, but which have multiple peaks, so we would like to approximate complexity with a function that increases steeply initially, then tails off as it approaches an asymptote. We have had good results when applying the following formula as an estimate of the solution quality after n iterations:

$$C(MF, n, S) = \frac{PMF}{1 + e^{\frac{-n}{\log_2 N^{d_{MF}}}}} \quad (4)$$

where d_{MF} is the dimensionality of the model family.

For example, a producer that knew it would have 20 iterations to interact with a consumer population could use this formula to predict its performance using two-part tariff. Two-part tariff has a precision of 0.90 for the problem described previously, and its dimensionality is 2. Let us assume that prices are within the range $[0,100]$. Plugging in these numbers, we find that $C(Twopart, 20, p^*) = 0.736$. So, in the 20th iteration, a producer using 2-part tariff would extract $\frac{0.736}{1.152} = 64\%$ of the available profit (in expectation). By integrating equation 4 over $[0,19]$, we get the total profit (per customer). A producer using 2-part tariff for 20 iterations can expect to earn a total undiscounted profit of 12.09, or 52% of the available profit. Similarly, a producer using block pricing would have $C(Block, 20, p^*) = 0.695$ and a cumulative profit of 11.79, so two-part tariff would be a better model. As the number of iterations increases, C will increase for each model and block pricing's higher precision will make it a more attractive choice.

Choosing a model

If the producer is to choose a schedule and then interact with a population for n iterations, it would use equation 4 to determine the profit at the current iteration and at n iterations. By integrating this equation, it can determine the expected profit for each model and choose the model with the highest expected aggregate profit.

But what if the producer wants to change schedules as it learns? In order to determine the optimal sequence

of models to use, a producer must consider not only the profit a model can allow it to earn, but also what data acquired in the context of that model allows it to learn about other models. In the most general case, a producer would have to consider all possible sequences of schedules. We avoid this problem by having the producer make a greedy estimate of each schedule's benefit; at each iteration, it chooses the schedule which it believes will yield the greatest cumulative profit. To make this effective, we make one simple assumption: the inputs of a model family at dimension d are a subset of the inputs of a model family at dimension $d + 1$. For example, linear pricing can be thought of as two-part tariff with the subscription fee set to 0. This allows a learning producer to perform a greedy search through the space of model families. The question to consider is how much is learned in the more complex model from data collected in the simpler model. Our solution is to weight the number of data points by the ratio of dimensionalities of the two models. That is, if a producer collected n data points from a model of dimensionality d , we would treat this as $n \frac{d}{d+k}$ data points for the purposes of evaluating the complexity of a model of dimensionality $d + k$. This reflects the fact that, by learning a d dimensional model, one is also learning a d -dimensional subspace of the larger $d + k$ -dimensional model.

Experimental Results

In order to test whether a producer using model updating does in fact outperform a producer using a static model, we conducted a set of experiments comparing the two sorts of producers. We created a set of producers that used static models (one for each price schedule), and had them learn the optimal prices for their particular schedule when interacting with an unknown consumer population having $w = 10$ and k drawn from $U[0, 0.7]$. Each producer used Amoeba (Press 1992), an implementation of the Nelder-Mead algorithm for nonlinear hillclimbing, to learn the optimal prices. We compared these results to a decision-theoretic producer that simultaneously maintained multiple models, which were also learned with Amoeba. A graph comparing the performance of an adaptive producer to static producers is shown in Figure 2.

As we can see from the graph, the decision-theoretic learner (denoted as 'Adaptive Pricing') is able to outperform most of the static schedules for the majority of the experiment. The schedules used by the adaptive learner are indicated at the top of the figure. It begins by trying a number of different schedules to collect data, and then settles on linear pricing for approximately 35 iterations. It then switches to two-part tariff for about 60 iterations before moving on to block pricing. Finally, at around 300 iterations, it switches to nonlinear pricing.

Also, we note that there is no static schedule that dominates adaptive pricing over the course of the entire experiment. This implies that if a producer is unsure of how long it will have to interact with a population, adaptive pricing is an effective approach. In addition, adaptive pricing performs particularly well in the 'intermediate' range of 10-200 iterations. In this range,

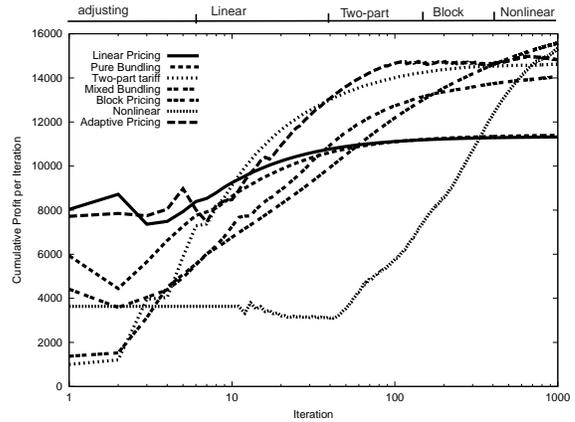


Figure 2: Cumulative profit per article, per period, per customer ($N = 10$) (averaged over 10 runs)

learning is able to play an important role. If a producer has only a few iterations to interact with a population, there is not enough time to learn much. Conversely, if a producer has a long time to interact with a given population, schedules can be learned exactly and long-run steady-state profits will dominate.

Current Research

We are also interested in extending this idea beyond price schedules to consumer preferences over types of articles. If a producer must also choose what articles to offer to a population and it has access to a (possibly incomplete) taxonomy describing categories of articles, it could employ this procedure, initially offering broad categories and refining its offerings to more specific categories as information is acquired. As with price schedules, the challenge is in applying knowledge acquired with one model to the updating of other models.

We would also like to be able to apply this technique to populations that learn or adapt continuously. A producer that noticed a change in the population could temporarily shift to a simpler model, tune this model to the new population, and then revert to a more complex schedule.

References

- Brooks, C. H.; Fay, S.; Das, R.; MacKie-Mason, J. K.; Kephart, J. O.; and Durfee, E. H. 1999. Automated strategy searches in an electronic goods market: Learning and complex price schedules. In *Proceedings of ACM EC-99*, 31–40.
- Brooks, C. H.; Durfee, E. H.; and Das, R. 2000. Price wars and niche discovery in an information economy. In *Proceedings of ACM Conference on Electronic Commerce (EC-00)*.
- Chuang, J. C.-I., and Sirbu, M. A. 1997. Network delivery of information goods: Optimal pricing of articles and subscriptions. In *Proceedings of the Conference on Economics of Digital Information and Intellectual Property*.
- Press, W. H. 1992. *Numerical Recipes*. Cambridge University Press.