The Gestalt in Graphs: Prediction Using Economic Networks

Vasant Dhar* Gal Oestreicher-Singer* Arun Sundararajan* Akhmed Umyarov*

Abstract

We define an economic network as a linked set of entities, where links are created by actual realizations of shared economic outcomes between entities. Such networks are becoming increasingly prevalent on the Internet, an example being the copurchase netwok on Amazon where entities are books and links designate which pairs were purchased simultaneously. Our dataset covers a diverse set of books spanning over 400 categories over a period of three years with a total of over 70 million observations. To our knowledge, this is the first large scale study showing that an economic network contains useful predictive information that is distributed in the network. We show that an economic network contains predictive information. Specifically, we demonstrate that an entity's future demand is more accurately predicted by combining its historical demand with that of its neighbors than by considering its demand alone. In other words, if you want to know what your state will be in the future, consider what is happening to your neighbors now. This result could apply to other economic networks where outcomes of sets of entities tend to be related.

1 Introduction

The increase in commercial and social interaction online have made electronic networks of different kinds increasingly prevalent. These networks may provide useful information that is not available if the elements of the networks are considered in isolation. One such network is a *social network* which describes relationships between individuals who are friends, colleagues, or trading partners (facebook, LinkedIn, MySpace, just to name a few). When people are "connected" to each other in a social network chances are that the individuals share some common interests or objectives and may respond similarly to certain external stimuli or even influence each other. In a recent book [3] even see happiness and health related issues as collective network oriented phenomena.

A different kind of network which has received less attention in the literature and popular media thus far but which we believe is central to graph-based predictive modeling, is an *economic network*. In economic networks or graphs, the links of the network are created by actual realizations of economic outcomes related to and relating the entities which are the nodes of the netwok. One widely occurring example is a copurchase network (i.e., a network that connects products based on shared purchasing patterns) often presented on electronic commerce sites. When items tend to be purchased concurrently, chances are that something in common affects their demand. This information can be very useful especially if what is causing the correlation is distinct from their observable characteristics like the author of a book or the category of a product. Indeed, collaborative filtering has been tremendously successful for Amazon which is rumored to derive roughly 20% of its sales through recommendations of products that have been "copurchased" in the past.

There has been considerable interest recently in data mining that is based on social networked data [6], [5] with fraud detection, marketing, and counterterrorism being some of the popular applications. [1] While there has been a lot of recent attention directed towards social networks, interactions between social entities are often complex, involving many types of interaction ranging from information sharing to recommendations. In contrast, online economic networks are relatively simple due to the "passive" nature of the interaction among the entities. Moreover, these networks may include valuable information that can be used to improve outcome predictions, while also making it possible to identify and measure the influence of links. This research asks whether economic networks contain any predictive information, where changes in the state of one entity result in subsequent changes in the state of entities linked to it. In other words, is the collective knowledge about the configuration or pattern of a set of entities linked by economic outcomes greater in value than the knowledge of each entity in isolation, or is there a form of "gestalt" associated with the economic graphs that will allow us to exploit their structure to build better predictive models?

*New York University. Stern School of Business.

[1] A more comprehensive survey is beyond the scope of this paper, although a bibliography of loosely related papers is available at *http://www.cs.purdue.edu/homes/neville/courses/icwsm09-tutorial.html*

If economic networks do contain useful predictive information, it is distributed among its entities and links, driven by phenomena such as preferences of consumers and similarities among products. A central advantage of using economic networks is that their existence eliminates the need to actually model these preferences and similarities explicitly, often a near-impossible undertaking. Consider a simple example from electronic commerce. Many large online retailers have tens of millions of consumers. Each of these consumers has a unique set of preferences and willingness-to-pay for the (possibly) millions of products that are sold by the retailer. In order to use data mining techniques for predicting optimal marketing choices, or for demand forecasting and planning, it is customary to create a coarse partition of these consumers along a small set of readily observable dimensions, such as gender, zip code, and age, and other behavioral profiles. The data mining task is to relate choices or actions to these dimensions with the reasoning that consumers who share common characteristics along these dimensions are likely to make similar choices.

The following economic analogy provides some insight into our conjecture that links created by economic outcomes have information of predictive value. Equilibrium prices are "determined" for a variety of products every day, and equilibrium demand levels are realized at posted prices, for example, at a variety of retail stores. While it is not possible to "reverse engineer" the actual preferences of decision makers or characteristics of products from these observed prices or demand levels, it is widely accepted that such prices contain aggregated summaries of these preferences. If one can relate these products to one another based on similarities in such economic outcomes and if there is sufficient gestalt associated with the network, the observed outcomes today for the neighborhood of a product may be good predictors of future outcomes for the product itself.

We investigate this possible relationship using a massive copurchase dataset gathered from Amazon.com. The data cover over 700,000 books over three years resulting in a total of over 70 million daily observations. In the past, [7] used a similar dataset to quantify the increased sales of books due to the visibility of this network. They found that the visible presence of links between complementary products triples, on average, the measured demand complementarity. In other words, during the period in which two products are linked, for an increase in the demand of one of them, the demand for the other increases three times more than it would if it were a simple complementary product with no visible online network link.

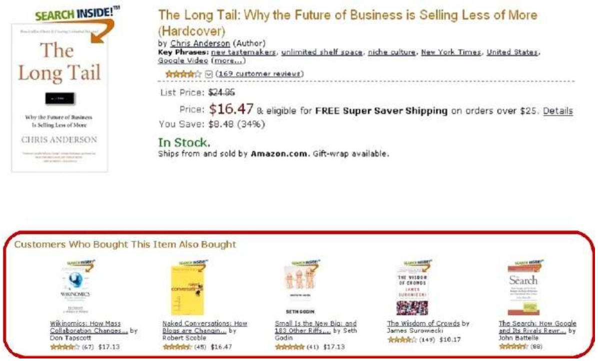
This prior result highlights the importance of links that are created or altered between two entities based on them sharing an economic outcome. Our current question which relates to outcome based links can thus be operationalized in the context of these network links which are created by a high fraction of copurchases. The interconnection between economic objects (agents, products) is not on account of their explicitly sharing one or more observable features or characteristics such as author, topic or genre. Such features are part of what we term a product's "intrinsic features", and are typically used as a basis for prediction in data mining. Our interest here is in demonstrating that an entity's future demand is more accurately predicted by combining its historical demand with that of its neighbors than by considering its demand alone, and this kind of improvement can be obtained even when using a relatively simple autoregressive model.

To summarize, we examine the predictive power of these copurchase links, asking whether the network data provided by the copurchase links helps predict changes in demand for books better than is achievable without consideration of the network. Our baseline model is a simple auto-regressive (AR) model [8] where demand in the next period is a linear combination of demands in previous periods. We also specify an AR model that includes information about the network as specified below. We vary each type of model by including different amounts of history in making the prediction. By comparing the results of the predictions between the two types of models we can assess whether information about the network has consistent predictive value.

2 Data

We use a large time series data set of recommendation networks for over 700,000 books sold on Amazon.com. Each product on Amazon.com has an associated webpage. These pages each have a set of copurchase links which are hyperlinks to the set of products that were copurchased most frequently with this product on Amazon.com. This set is listed under the title "Customers who bought this also bought:". An example of copurchase links is illustrated in Figure 1.

The copurchase network is a directed graph in which nodes correspond to products, and edges to directed copurchase links. We collect data about this graph using a Java-based crawler, which starts from a popular book and follows the copurchase links using a depth-first algorithm. At each page, the crawler gathers and records information for the book whose webpage it is on, as well as the copurchase links on that page, and terminates when the entire connected component of the graph is collected. This is repeated daily. A sample part of the graph is illustrated in Figure 2 and the



SEARCH INSIDE!™

The Long Tail: Why the Future of Business is Selling Less of More (Hardcover)
by Chris Anderson (Author)
Key Phrases: new tastemakers, unlimited shelf space, niche culture, New York Times, United States, Google Video (more...)
(169 customer reviews)

List Price: ~~$24.95~~
Price: $16.47 & eligible for **FREE Super Saver Shipping** on orders over $25. Details
You Save: $8.48 (34%)

In Stock.
Ships from and sold by **Amazon.com.** Gift-wrap available.

Customers Who Bought This Item Also Bought

Wikinomics: How Mass Collaboration Changes... by Don Tapscott (67) $17.13

Naked Conversations: How Blogs are Changin... by Robert Scoble (45) $16.47

Small Is the New Big: and 183 Other Riffs,... by Seth Godin (41) $17.13

The Wisdom of Crowds by James Surowiecki (149) $10.17

The Search: How Google and Its Rivals Rewr... by John Battelle (88)

Figure 1: An Example of Copurchase Links

corresponding larger segment of a copurchase network is shown in Figure3. The algorithm used for data gathering is provided in the Appendix A.1

We have chosen to focus on books because they are in the product category with by far the largest number of individual titles, whose product set is relatively stable (compared to electronics, for instance), and the network data are observable.

The data collection began in August 2005 and is currently ongoing. The graph is traversed every day. The following data is available for each book on the copurchase graph, for each day:

- **ASIN:** a unique serial number given to each book by Amazon.com. Different editions and different versions have different ASIN numbers.
- **List Price:** The publisher's suggested price.
- **Sale Price:** The price on the Amazon.com website that day.
- **Copurchases:** ASINs of the books that appear as its copurchases.
- **SalesRank:** The sales rank is a number associated with each product on Amazon.com, which measures its demand of relative to other products. *The lower the number is, the higher the sales of that particular product.*
- **Category Affiliation:** Amazon.com uses a hierarchy of categories to classify its books. Thus, each book is associated with one or more hierarchical lists of categories, starting with the most general category affiliation, and ending with the most specific one. For example: *Subjects, Business & Investing, Biographies & Primers, Company Profiles*
- **Author:** The name of the book authors.

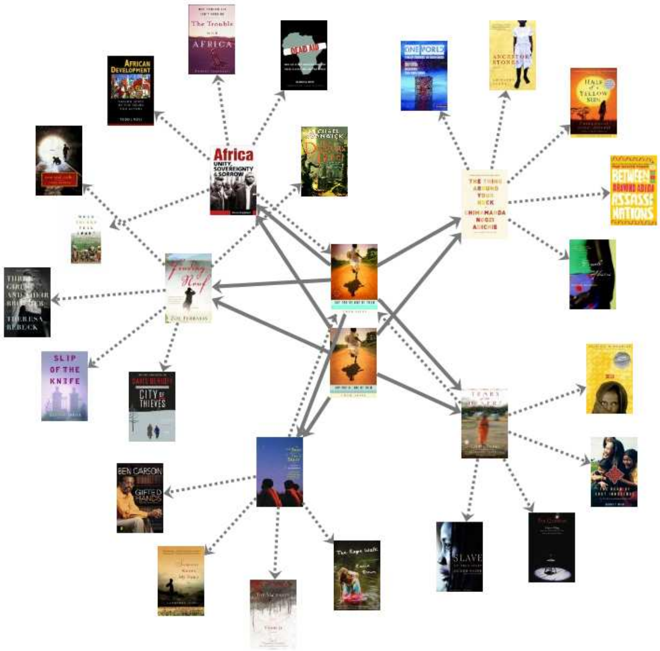


Figure 2: Small segment of a copurchase network

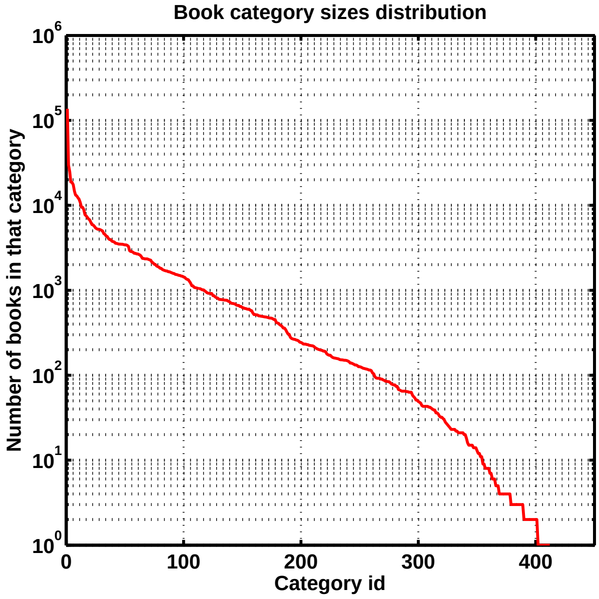


Figure 4: Distribution of category sizes

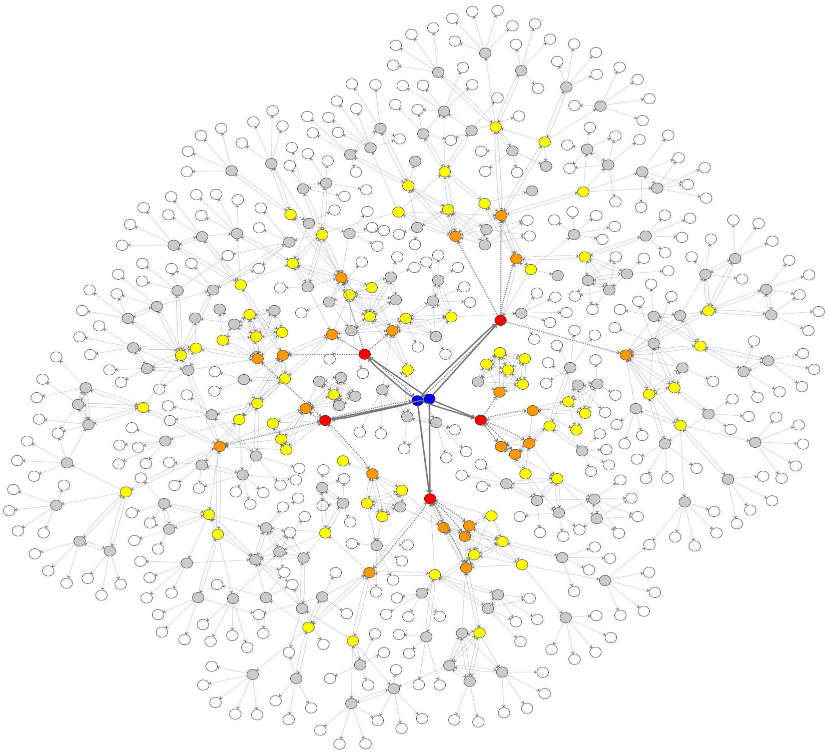


Figure 3: Larger subset of the copurchase network

- **Publisher:** The name of the book publisher.

- **Publication date:** The date of publication of the book (by that publisher).

An additional script collects the demand information for all books on the graph every 3 hours for the 24-hour period following the collection of the graph.

The number of books per category varies considerably as shown in Figure 4. For the purposes of this study, however, we do not distinguish between categories, and attempt to build a general model that treats them equivalently. We return to this issue in the Discussion section.

3 The sales rank prediction task

The dependent variable that we set up for our prediction task is the salesrank of a book in the next period, since demand data are not directly provided for us.

For starters, we consider only the non network features, namely, those that are observable and specific to the product, or what a typical predictive modeling exercise might consider as the basis for its features. To build a predictive model, one would typically construct and select a number of features that have some correlation to sales. These might be things like changes in average daily SalesRank for a number of weeks, percentage of days that the average daily SalesRank increased on a daily basis during the last N days, list price of a book, weekly average sale price in the last N days, and so on. In addition, there could be seasonal variables, especially for certain genres of books.

Recall that our objective in this paper is not to build the best predictive model or even a good one. Rather, it is to use a simple model that enables us to assess whether the network contains any useful predictive information. Accordingly, we limit our model structure to a simple autoregressive AR(N) model and consider only the past sales of the book upto some period in the past as independent variables. We also consider only those books that are “immediate neighbors” in the copurchase network, namely, those that have an explicit recommender link in Amazon.

4 The models

In order to present a reasonable predictive model for salesrank, we first consider the properties of this observed variable.

First, today’s salesrank of a particular item is highly correlated with the item’s yesterday’s salesrank. For the most part, most items have an established level

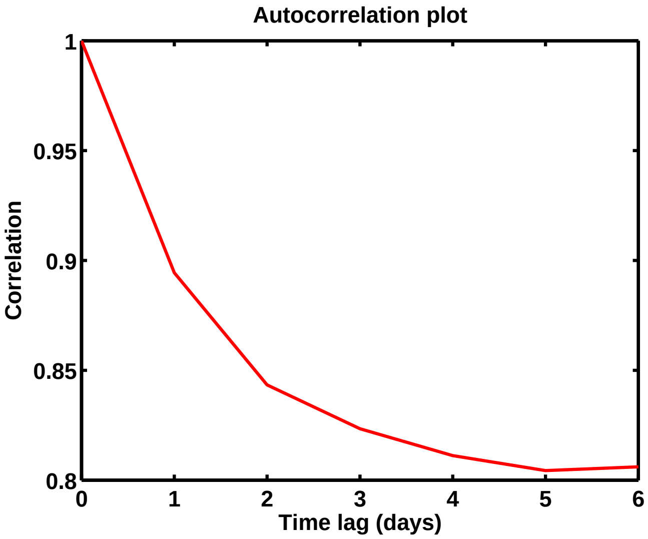


Figure 5: SalesRank autocorrelation plot

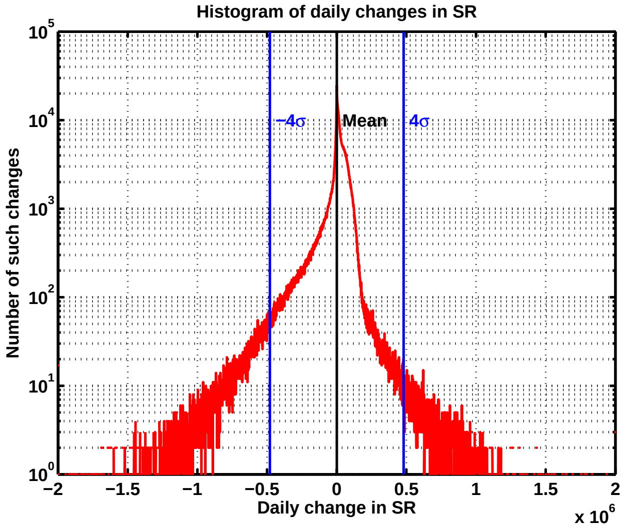


Figure 6: Changes in Salesrank

of popularity that does not vary much in the short term. This property is also reflected in the Figure 5, which shows the persistence of autocorrelation in salesrank. Figure 5 shows the correlation between today's salesrank and the salesrank several days ago. For example, the value 1 on the x-axis corresponds to the estimated correlation between today's salesrank and yesterday's salesrank, the value of 2 on the x axis corresponds to correlation between today's salesrank and the day before yesterday's salesrank and so on.

Second, we take into account that a drop in salesrank from #1 to #20 is conceptually very different from drop of salesrank from #1001 to #1020. In the first case, the underlying demand is likely have changed on a large scale, while in the second case, the change in underlying item demand is likely to be very little if any. Therefore, the change of +20 in raw salesrank only has meaning depending on the level of salesrank.

Note that no such argument applies directly to logarithm of salesrank: a drop in demand that lies behind the drop in salesrank from #1 to #20 is comparable to drop in demand when salesrank drops from #100 to #2000. This intuition was indeed experimentally confirmed in multiple studies of relationships between salesrank and demand such as [4], [2]. We make use of this relationship as explained later in this section.

Finally, Figure 6 shows that the changes in raw salesrank values tend to exhibit a very non-symmetric distribution, since the nature of salesrank variable is such that if a single book rockets up in popularity, most of other books get shifted down, by a little.

In contrast, the changes in logarithms of salesrank shown in Figure 7 exhibit more symmetric distribution. For the reasons mentioned above, we considered that for a salesrank predictive model it is more appropriate:

1. to use the log of salesrank as the dependent variable, instead of raw salesrank value
2. to use AR-style model that exhibits similar autocorrelation patterns [8] to the one shown in Figure 5.

Therefore, we used the following 3 functional forms of the AR model as the predictive models to be compared:

1. Model 1. *Naive model.* This model predicts the salesrank today with yesterday's salesrank as follows:

   $$\log(SR_0) = \log(SR_1) + \varepsilon, \quad E[\varepsilon] = 0, \, \mathrm{Var}[\varepsilon] = \sigma^2$$

   where SR_0 is unobserved salesrank today, SR_1 is observed salesrank yesterday and ε is an unobserved error term with unknown variance σ^2

2. Model 2. *The baseline autoregressive (AR) model.* This model predicts the salesrank for an item today from the observations of salesrank for that item for N previous days as follows:

   $$\log(SR_0) = \alpha_0 + \sum_{i=1}^{N} \alpha_i \log(SR_i) + \varepsilon$$

   where SR_0 is unobserved salesrank today, SR_i is observed salesrank i days ago and ε is an unobserved error term as described above.

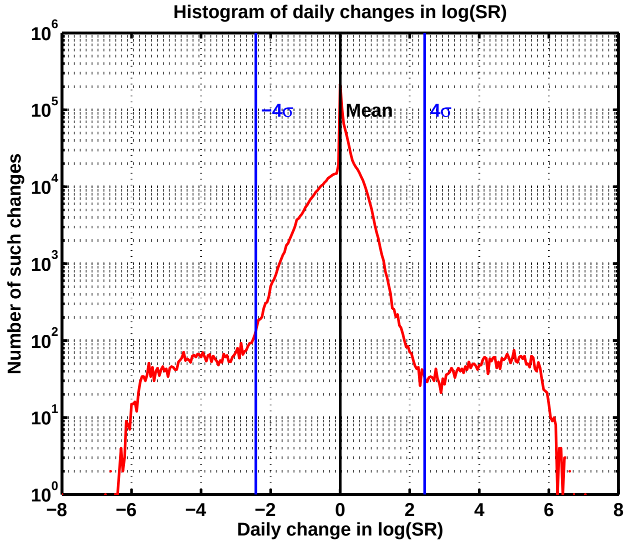


Figure 7: Changes in Log(Salesrank)

3. Model 3. *Network effects autoregressive model.* This model augments the baseline AR-model described above with the data of salesrank for network neighbors for the past M days:

$$\log(SR_0) = \alpha_0 + \sum_{i=1}^{N} \alpha_i \log(SR_i) + \sum_{j=1}^{M} \beta_j \log(N_j) + \varepsilon$$

where all SR_i and ε are as described above and N_j is the "averaged" salesrank for network neighbors j days ago. This calculation of the average salesrank for network neighbors is described in detail in Appendix A.2.

As is usually the case in AR models, we assume that the noise term ε is stationary and does not change its properties over time.

In this study, we also assume that each item salesrank follows the same underlying AR-process and therefore, the model coefficients α_i, β_j are the same for each book and each time period. While it is possible to have the coefficients be book specific, such a model would require large amounts of data for each book to be realistic. Assuming a general model limits the possibility of overfitting and keeps the evaluation simple.

4.1 Model estimation We estimate the coefficients $\{\alpha_i\}$ and $\{\beta_j\}$ of the models 2 and 3 using ordinary least squares estimator. This approach is standard and well-known for AR models. Below we demonstrate it for model 3 only, but exactly same logic applies for model 2 as well.

Assume that $\log(SR_0)$ is unobserved random variable which we denote as y for notational convenience. Assume also $\log(SR_1), \ldots, \log(SR_N)$ and $\log(N_1), \ldots, \log(N_M)$ are observed values and we put them into a column-vector x. Denote the vector of all unknown coefficients $(\alpha_1, \ldots, \alpha_N, \beta_1, \ldots, \beta_M)$ as vector γ. Then our model can be represented as

$$y = x'\gamma + \varepsilon, \quad E[\varepsilon] = 0, \ \mathrm{Var}[\varepsilon] = \sigma^2$$

that is a standard linear regression model.

More specifically, assuming that all observed values of y are stacked into the column vector Y and all the corresponding observed values of x' are stacked into the matrix X, the unknown coefficient γ of such model can be estimated consistently and efficiently with ordinary least squares method [8] as follows

$$\hat{\gamma} = (X'X)^{-1}X'Y$$

More detailed treatment of this approach can be obtained from [8].

5 Results

We partitioned the data into a training and a test set using a cutoff date method. We chose the cut-off date of April 20, 2007 since it creates a training set of 80 percent and a test set of 20 percent. According to this cut-off date procedure, every observation that is dated before April 20, 2007 falls into the training set and every observation that is dated after April 20, 2007 falls into the test set.

This was a stringent test in that several books appeared only in the test set and never appeared in the training set. We acknowledge that there are other methods for performing the in and out of sample analysis, which we intend to conduct in subsequent research that we consider in the Discussion section, but our preliminary analysis on a couple of categories suggests that the results are very stable across different data splits.

The results of the regression are summarized in Figure 8. The y axis in the figure is the performance variable, namely the mean squared error. The x axis is the number of days N and M of history included into the model[2].

The results are interesting in several ways. First of all, the improvement in error is consistent, roughly around ≈ 0.5 % for the most complex models. The network provides a higher percentage improvement when a shorter history is used. How significant an improvement is the improvement of 0.5 percent? It is well known

[2] Here we use $N = M$.

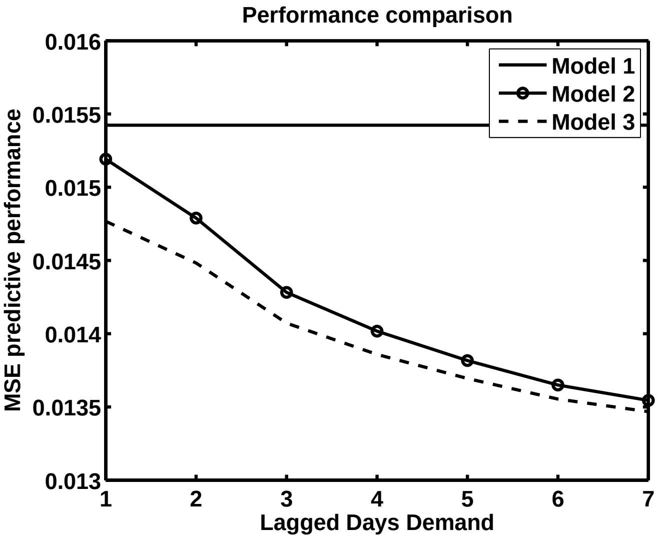


Figure 8: Performance results

that for problems with a lot of noise, most models, and especially AR models, tend to produce forecasts that are very close to the mean. In other words, given the noise in the problem, it is very difficult for a forecasting model to predict large values for the dependent variable since these often turn out to be wrong whereas the model minimizes error by predicting values very close to the mean. Accordingly, the majority of the error is contributed by the large actual values of the dependent variable, so "improvements" in overall error tend to be very small. In this respect, an error reduction of 0.5 % across the entire dataset of more than 70 million observations could be a very significant improvement. Indeed, the standard errors for the estimates are very small, of the order of 10^{-5}.

In addition to looking for improvement in error, we considered it more relevant to compare errors of the two models across different levels of model complexity. This was the motivation for analyzing the errors by considering varying histories of predictors. The results here show that the network model consistently provides a lower error rate than the one without the network.

6 Discussion and Future Work

In previous research reported in the literature by [7], it was shown that the impacts of visible network links could be econometrically identified and that their economic impact was significant. In this study, we set out to answer a different question, namely, whether changes in demand can be predicted more accurately using network information than without it. The results show this is in fact the case. To the best of our knowledge, we provide the first evidence using a large scale study on the existence of predictive information contained in the structure of economic networks.

As a first study that uses economic networks for predictive modeling, we restricted our attention to the simplest possible "neighborhood" of a product, its immediate in-neighbors. Our current research aims to extend this neighborhood to include more distant neighbors in constructing the network related features. A useful summary of the level of influence that a graph has on each of its nodes is embodied in the PageRank measure of centrality [1]. We are constructing a similar measure of centrality that additionally, weights nodes in a neighborhood by their demand. We also believe that large changes in an appropriately defined measure of graph centrality will be good lagging indicators of demand shifts. Such changes are likely to be associated with a large shift of the product within the graph, which in turn would create a new neighborhood and new demand influences for the product whose measurable impact may take time to manifest. Testing the latter conjecture remains a promising line of future inquiry.

While our intuition at the outset of the project was that the economic network contains useful predictive information about sets of linked entities, it was not obvious that these would manifest as lagged effects. In contrast, if for example, new information is reflected instantly in related entities, we would expect changes in the state of entities to be concurrent. The result suggests that this is not the case, and that additional useful information is distributed in the neighborhood of a product. Furthermore, such information can be aggregated for predictive purposes. A natural next step in this direction would be to associate varying levels of lag influence with nodes that are different distances from the product in question. For example, one might expect the influence of a distant neighbor to take longer to measurably affect the economic outcomes we observe.

Having validated our initial central conjecture, namely that economic network contains predictive information, it seems natural for future research to consider the use of machine learning methods as an alternative the AR model to build more accurate predictive models. Machine learning methods can deal with problems with high levels of noise by discovering "local" models that can be applied to different partitioning of the data. These localized models can make more aggressive forecasts for specific books in contrast to the AR model that is very conservative in its predictions, with small deviations from the mean. It is also worth considering additional ways to validate the results. One method would be to hold the last few observations of every book for testing. In this way, the model would be built and tested using time series data on every book. We would expect that the results with this partitioning to be better compared to the method we used where many books don't even appear in the training sample.

We expect that there will be numerous new economic networks that become observable to firms and researchers over the next years. The tight association of the links in these networks with economic outcomes of interest makes them especially attractive as a basis for predictive modeling. These networks are the natural place to start when looking for ways to expand beyond product-centric features. This is because they do not just have some subset of the vast amount of more "global" information that a predictive modeler would like to know to attain better accuracy but also automatically end up zeroing in on and containing the small fraction of such information that is actually useful, on account of having links generated by events associated with the very outcomes one is interested in predicting.

References

[1] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[2] E. Brynjolfsson, Y. Hu, and M. D. Smith. Consumer surplus in the digital economy: Estimating the value of increased product variety at online booksellers. *Management Science*, pages 1580–1596, 2003.

[3] N. A. Christakis and J. H. Fowler. *Connected: The Surprising Power of Our Social Networks and How They Shape Our Lives.* Little, Brown and Company, 2009.

[4] A. Goolsbee and J. Chevalier. Measuring prices and price competition online: Amazon and Barnes and Noble. 2003.

[5] S. Hill, F. Provost, and C. Volinsky. Network-based marketing: Identifying likely adopters via consumer networks. *Statistical Science*, 21(2):256–276, 2006.

[6] J. Kleinberg. Challenges in mining social network data: processes, privacy, and paradoxes. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 4–5. ACM New York, NY, USA, 2007.

[7] G. Oestreicher-Singer and A. Sundararajan. The Visible Hand of Social Networks in Electronic Markets. *SSRN.*

[8] R. Tsay. *Analysis of financial time series.* Wiley-Interscience, 2005.

A Appendix

A.1 Algorithm for Data Collection

We use two programs for the collection of my data. The first collects graph information and the second collects sales rank information. Both use the Amazon.com's XML data service. This service is part of the Amazon Web Services, which provides developers with direct access to Amazon's platform and databases.

Graph Collection: The program which collects the graph starts at a popular book. It then traverses the copurchase network using a depth-first search. Intuitively, in a depth-first search one starts at the root (in our case, the one popular book chosen) and traverses the graph as far as possible along each branch before backtracking. At each page, the crawler gathers and records information for the book whose webpage it is on, as well as the copurchase links on that page. The ASINs of the copurchase links are entered to a LIFO stack. If the algorithm finds it is on a page of a product that it has visited already, it backtracks and returns to the most recent product it hadn't finished exploring. The program terminates when the entire connected component of the graph is collected.

For example, in the graph on Figure 9, the nodes are numbered in the order in which the crawler with traverse the graph. In this case, the collection starts at node 1. Its copurchase links are nodes 2, 6, 7. Therefore, those numbers are added to a LIFO stack. The script will then proceed to node 2, whose copurchases are nodes 3, 4, 5 and thus, those numbers will be added to the LIFO stack, which will now include: 3, 4, 5, 6, 7. The script will continue to node 3. Since there are no copurchase links to that node, it will move to node 4. In the same way, the script will collect data about node 5, node 6 and node 7.

Since node 7 has copurchase links – nodes 8 and 9, they will be added to the stack. After visiting nodes 8, 9 and 10, the data collection will terminate. As can be seen, the script only stops once it has collected information about the entire connected component.

The collection of the entire connected component on Amazon.com takes between four and five hours. The script is run each day at midnight.

Sales Rank Collection: A second program collects the demand information for all books on the graph every 3 hours for the 24-hour period following the collection of the graph. This script collects the Sale Rank of all the books which ever appeared on the graph. Therefore, it follows the sales of books that are no longer on the graph as well.

A.2 Converting Sales Ranks to Demand

SalesRank is a number associated with each product on Amazon.com, which measures its demand relative to the other products sold on Amazon.com. The lower the number is, the higher the sales of that particular product. The sales rank of a book is updated each hour to reflect recent and historical sales of every item sold on

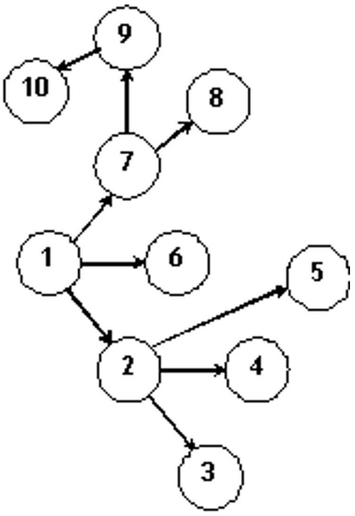


Figure 9: Node traversal

Amazon.com.

A formula to convert SalesRank information into demand information was first introduced by [4]. Their goal was to estimate demand elasticity. Their approach was based on making an assumption about the probability distribution of book sales, and then fitting some demand data to this distribution. They choose the standard distributional assumption for this type of rank data, which is the Pareto distribution (i.e., a power law). In the Pareto distribution, the probability that an observation's value, exceeds some level, S, is an exponential function

$$\text{(A.1)} \qquad \Pr(s > S) = \left(\frac{k}{S}\right)^{\theta},$$

where k and θ are the parameters of the distribution. The more important parameter is θ, the shape parameter that indicates the relative frequency of large observations. If θ is 2, for example, the probability of an observation decreases in the square of the size of the observation. With a value of 1, it decreases linearly.

For a given book, the number of books that have sales greater than that book is just one less than the books' rank. Therefore, the fraction of all books that have sales greater than a particular book is just

$$[\text{SalesRank} - 1]/\text{TotalNumberOfBooks}$$

If there are a sufficient number of books to eliminate the approximation introduced by discreteness, then one can replace the equation above with:

$$\text{(A.2)} \quad \frac{[\text{SalesRank} - 1]}{\text{TotalNumberOfBooks}} = \left(\frac{k}{\text{Demand}(j)}\right)^{\theta}$$

Taking logs, and substituting θ with $-1/b$, this translate ranks into sales according to

$$\text{(A.3)} \quad \log[\text{Demand}(j)] = a + b \log[\text{SalesRank}(j)]$$

The parameters a and b were estimated by Goolsbee and Chevalier using a couple of parallel methods: using data from the Wall Street Journal book sales index, which gives the actual quantity sold; using sales information given by a publisher, who sells on Amazon.com; conducting an experiment, buying copies of books with a steady salesrank.

In a later study, [2], used data provided by a publisher selling on Amazon.com to conduct a more robust estimation of the parameters of the formula. They estimate the parameters as: $a = 10.526$, $b = -0.871$.

To avoid hour-of-the-day effect, we use average daily salesrank in our model estimations. Since our crawler collects salesrank every 3 hours, we required averaging over the day. However, there is no straightforward meaningful way of averaging salesranks. Therefore, we employed the following procedure to compute the "averaged" salesrank:

1. We used the above mentioned conversion equation to transform the hourly salesrank data into 3-hourly demand data.

2. We then averaged these 3-hourly demand into a daily average demand

3. We converted the average daily demand back into an artificial "average salesrank" using the above conversion formula.

This computational formula makes more sense than simple averaging of salesranks, since the underlying variable for averaging in this formula is demand and averaging 3-hourly demand has a meaningful interpretation of being an average demand for the item throughout the day. Therefore, the generated artificial "average salesrank" has a meaning of being the salesrank of an item that would have had this average demand observed constantly throughout the day.