Leveraging Bidder Behavior to Identify Categories of Substitutable and Complementary Goods on eBay

A thesis presented
by
Robert Kang-Xing Jin
to
Computer Science
in partial fulfillment of the honors requirements
for the degree of
Bachelor of Arts
Harvard College
Cambridge, Massachusetts

April 4, 2006

Contents

Chapter 1

Introduction

Online auctions continue to grow at a fast pace. For example, at the end of 2005, the leading online auction site eBay.com had 180.6 million registered users, a 33 percent increase over the previous year. These users listed a total of 546.5 million items in the quarter; the total value of successfully closed items was $12.0 billion [4].

The increasing prominence of online auctions has sparked interest in efforts to better understand their economic properties. Much theoretical work has been done on designing online auction mechanisms with desirable properties such as revenue-maximization [12]. At the same time, the scale of the online auction market provides a uniquely rich data set for empirical studies. In addition to more direct applications, such as fraud detection [24], empirical studies also inform theory—for example, mechanism designers often make strong assumptions about bidder behavior, and empirical studies can provide realistic guidelines for these priors.

In this thesis, we explore, using data collected from eBay, an empirical problem in the online auction space: given a large auction market with millions of widely varying items—eBay, for example, sells antiques, cars, real estate, and electronics, among others—what is a scalable way to organize these items into categories? Online auctions rely on being able to match buyers and sellers, and it is important that they have a good categorization system that makes items easy to find. One study found that the presence of product navigation and categorization features in large e-commerce websites has a significant positive effect on monthly sales [10], and another study attributes eBay's success in part to its dynamic categorization scheme [25]. The site initially had only a few item categories; the taxonomic hierarchy was expanded dynamically as the site grew.

The categorization problem is interesting because the size of the market imposes a requirement for scalability. Thus, any method needs to be largely automated and cannot rely heavily on expert knowledge. While generating high-level categories, e.g., cars vs. electronics, might be relatively easy, the method must also be able to generate subcategories at the level of substitutable goods, e.g., one model of digital camera vs. another. After all, a bidder interested in a specific model of digital camera would probably like to be able to browse a subcategory for that specific model rather than having to browse all digital cameras. Indeed, items listed deeper in the eBay taxonomic category tree tend to attract more bidder traffic [7]. Consistent with this finding, some have argued for "surgical search" functionality that allows users on e-commerce websites to search low-level categories by category-specific parameters [11]. For example, in a monitor category, such parameters might include size, model, and brand.

Identifying substitutable goods at lower levels in the category hierarchy without resorting to specific knowledge about the goods—the substitutes problem—is the main problem that we specifically address in this thesis. The substitutes problem is a subproblem of the more general categorization problem discussed above. In our setting, "identification" entails being able to separate auctions into different groups of substitutable goods and also being able to extract representative keywords that distinguish the different groups from each other. For example, we would like to be able to divide a set of auctions for monitors into groups defined by relevant attributes such as size, model, and/or brand; these attributes could also serve as category-specific parameters for surgical search.

After addressing the substitutes problem, we extend our solution to a related problem, which we term the complements problem—identifying goods that bidders tend to buy together. We provide a more detailed explanation of our definitions for substitutes and complements in Chapter 3. One potential application for complements detection could be in automated "recommender systems" that suggest goods complementary to previous user purchases.

In the remainder of this introductory chapter, we review related literature, highlight our main results, and conclude with an outline of the rest of the thesis.

1.1 Related Work

The empirical study of online auctions is a relatively new field, and most of the work has been done in the past five years. This fact is not too surprising, since online auction websites have only reached prominence recently. We focus on the results of empirical studies and

leave technical details for later.

To our knowledge, no work has directly addressed either the substitutes problem or the complements problem for eBay. The closest related work is a study of bidder communities on eBay [20], which is the first study of communities in a network generated from an online auction site.[1] The authors logged all data over a 12-day period on the German eBay.de website. They then generated a network with bidders as nodes and edges drawn between any two bidders who bid in the same auction. Next, they applied a community detection algorithm to the network. They found 7 "major" communities and noted that these communities tended to correspond to auctions in specific eBay-defined high-level goods categories. For example, one community consisted of bidders who primarily participated in the Toy Models and Toy categories. From this data, they concluded that bidders tend to limit their activities to general categories of goods. This finding suggests that leveraging bidder activity information might be a good way to determine substitutable goods, especially if bidders also form communities at a lower substitutes level in the network. The approach that we propose to solve the substitutes problem makes use of this insight.

1.2 Main Results and Contributions

In this section, we summarize the main results contained in this thesis and highlight our contributions. The most significant results are found in our solution to the substitutes problem.

1.2.1 The Substitutes Problem

We propose a novel method for automatically identifying substitutable goods on eBay. We apply our method to data taken from eBay and demonstrate that it accomplishes its goal.

Our solution to the problem consists of three parts: (1) generating a network with auctions as nodes and edges drawn between auctions with shared bidders; (2) applying a community detection algorithm to the largest maximally connected component (MCC)[2] of the network; and (3) characterizing the communities found by applying a keyword extraction algorithm. The communities of nodes in the auction network correspond to communities of substitutable goods, and the keywords extracted could be potentially used as category headings or search parameters.

[1] Personal communication, M. Newman.

[2] We define an MCC as a connected subgraph where adding any node will result in it no longer being connected. The largest MCC is the MCC with the most nodes.

Aside from the overall solution itself, the main contributions of this work lie in the first part of our solution, the network generation. In this part, we have three main contributions. First, we apply, for the first time in the literature, community detection algorithms to an auction network with edges defined by some measure of shared bidder behavior. Second, we propose a natural way of weighting edges and show that edge weighting results in qualitatively better community and keyword results. Third, we propose methods of filtering edges to increase the substitute community structure of the network and demonstrate that these methods are effective. We show that our methods of edge filtering increase modularity (Q), a metric commonly used to assess community structure in a network [2, 15, 16, 14], and result in qualitatively better community and keyword results. We also show that we can filter a large percentage of edges before we start losing many nodes in the largest MCC, which is a property that a network should have in order for edge-filtering methods to be useful. This property is important because community detection algorithms can only be applied to connected graphs. If the largest MCC is only a small subset of the full network, then potential communities of goods might not be found via community detection.

For the second part of our solution, we implement an existing community detection algorithm [2]. One interesting result is that the networks we generate have extremely strong community structure. One of our networks had a modularity score higher than the highest known value for a real-world network in the literature [2].

For the third part of our solution, we propose a simple keyword extraction algorithm. We show that the keywords extracted from communities in edge-filtered, weighted networks correspond to reasonable categories of substitutable goods.

1.2.2 The Complements Problem

We propose a novel method for automatically identifying complement relationships between communities of goods on eBay. The complements problem is harder than the substitutes problem because there is less data available—as we shall see, relatively few bidders make complement-type purchases. In addition, the complements problem requires a solution to the substitutes problem, since one needs to identify the item types of interest before being able to assess complementary relationships. For example, if we are given a data set containing cameras and memory cards and want to assess complementary relationships between different types of cameras and different types of memory cards, we would need to first determine what the different types of cameras and memory cards are.

We examine two versions of the complements problem: (1) detecting complements if the substitute communities of interest are known beforehand, and (2) detecting complements if

the substitute communities of interest are not known beforehand.

To solve the first problem, we propose a class of metrics that measure the strength of complementarity between any two communities of goods. We evaluate these metrics on a data set of digital cameras and memory cards and demonstrate that the metrics can detect complementary relationships between models of cameras and the specific memory card formats used by those cameras.

To solve the second problem, we propose a three-step solution. The first two steps use a modified version of the community detection algorithm to automatically determine the substitutes communities of interest and group them into larger supercommunities defined by complementarity—that is, substitutes communities with strong complementary relationships will tend to be grouped together. Once these communities have been found, we apply our solution to the first version of the complements problem as discussed above. We evaluate our solution on the same data set and present preliminary results suggesting that it is effective.

1.3 Outline

In Chapter 2, we give an overview of our data set and collection methodology. Chapters 3 and 4 relate to the substitutes problem. In Chapter 3, we give a more specific definition of the substitutes problem and motivation for our approach to solving it. We then discuss the methods for each of the three parts of our solution. In Chapter 4, we discuss the results from applying our solution to the data set. In Chapter 5, we discuss the methods and results for our solution to the complements problem. In our concluding chapter, Chapter 6, we discuss potential extensions to our work, including examining how network structure changes over time and an application to hierarchical categorization.

Chapter 2

The Data

In this chapter, we give an overview of the data sets used in the thesis. We begin with a description of our collection methodology and then discuss the two data sets we used for the substitutes problem. Descriptions for the additional data sets we used for the complements problem can be found in Chapter 5. We then present statistics for the number of auctions bid on per bidder and the number of auctions won per bidder. Knowing these statistics will be useful for understanding some of the decisions later on in the thesis.

2.1 Collection Methodology

We collected data by searching closed listings on eBay.com. The harvesting scripts were based on those written by Jin et al. [8] and Roth et al. [21]. We used Perl scripts to scrape the data from the web pages and then stored the data in a Mysql database. For each auction, we collected the information specified in Table 2.1. We also collected all the bidder information from the history page associated with each auction, as specified in Table 2.2.

2.2 Data Sets

We collected data from two categories of goods. The first set (Canon) contains all auctions matching "Canon" in the Digital Cameras category over a period from January 10, 2006 to January 25, 2006. The second set (LCD) contains all auctions matching "LCD" in the Monitors and Projectors category over a period from Nov. 29, 2005 to Dec. 14, 2005. The sets were chosen because they are reasonably sized markets where there might be natural substitutes (specific models of cameras and specific sizes, brands, or models of LCDs).

Field	Description
aucname	The title field of the auction.
id	The eBay unique ID for the auction.
sellername	The name of the seller.
type	The type of the auction (Buy it Now or Standard).
reserve	The reserve price of the auction (if applicable).
sold	If the auction sold.
highbid	The high bid in the auction.
starttime	The start time of the auction.
endtime	The end time of the auction.

Table 2.1: Data Fields Collected for Each Auction Page.

Field	Description
bidname	The name of the bidder.
time	The time the bid was placed.
value	The value of the bid.

Table 2.2: Data Fields Collected for Each Bid History Page.

The Canon set consisted of 6717 auctions. 4308 of the auctions had at least one bidder (64%). 4107 (61%) of the auctions sold; the remainder may have either failed to meet a reserve price or have been cancelled by the seller. 3206 (48%) of the auctions did not have a buy-it-now option and 569 (8%) had a reserve price.

The LCD set consisted of 11782 auctions. 8288 of the auctions (70%) had at least one bidder. 7990 auctions (68%) sold. 6877 (58%) of the auctions did not have a buy-it-now option and 882 (7%) had a reserve price.

2.3 Bidder Statistics

In this section, we present some statistics describing bidder activity in each of our two data sets. Our approach to solving the substitutes problem relies on assumptions about bidder behavior, so it is important to have a basic understanding of how bidders behave. For the Canon set, 12759 unique bidders placed a total of 51648 bids. For the LCD set, 23801 unique bidders placed a total of 93912 bids.

2.3.1 Auctions Bid On Per Bidder

We examined the number of distinct auctions that each unique bidder participated in. Similar to Yang et al. [26], we constructed an undirected, unweighted bipartite graph with bidders and auctions as nodes. An edge was drawn between a bidder node and an auction node if the bidder placed a bid in that auction. We found, similar to Yang et al. [26], that for both data sets the degree distribution for bidders appeared to follow a power law (Canon: $y = 0.90x^{-2.54}$,$R^2 = 0.95$; LCD: $y = 0.42x^{-2.23}$, $R^2 = 0.91$).[1] In the Canon market, 8453 of the 12759 bidders (66%) participated in only one auction, and 2065 (16%) participated in only two. In the LCD market, 15650 of the 23801 bidders (66%) participated in only one auction, and 3883 (16%) participated in only two.

Graphs of the distributions for the Canon data set and LCD data set are shown in Figure 2.1. A small number of bidders thus account for a disproportionate amount of bidding activity. The maximum numbers of auctions participated in by a single bidder were 61 and 171 for the Canon and LCD markets, respectively. The bidder who participated in 171 auctions in the LCD market averaged more than 12 per day.

Power law distributions have been found in a variety of settings both in computer science (linking patterns in the World Wide Web) and elsewhere, leading some to propose generative models that result in these distributions [13]. One such model involves preferential attachment—that is, new nodes tend to link to the more highly linked existing nodes [1]. In the case of the bidder-auction network, however, a generative model makes less sense. The distribution is likely due to a property of bidder behavior—namely, that most bidders are interested in winning only one item (see next section) and thus place bids in an extremely limited subset of auctions.[2]

2.3.2 Auctions Won Per Bidder

We also examined the number of auctions won by each unique bidder. We found that the vast majority of bidders won only one item. For the Canon set, 4107 auctions were sold to 3821 unique bidders. Of these bidders, 3638 (95%) won only one auction. For the LCD set, 7990 auctions were sold to 7227 bidders. Of these bidders, 6708 bidders (93%) won only one auction. The distribution of the number of auctions won per bidder also appeared to

[1]Here R^2 is the square of the correlation coefficient and measures the linearity of the data. An R^2 of 1 corresponds to perfect linearity. A power law distribution is characterized by linearity on a log-log scale.

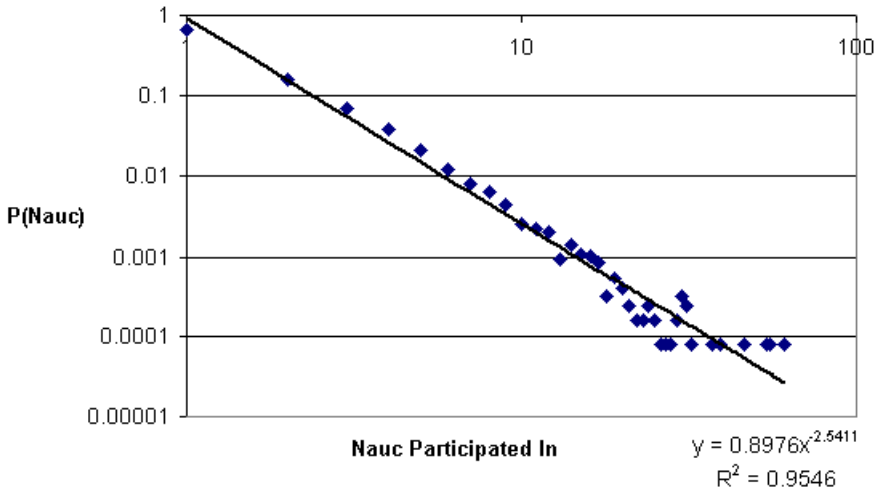
[2]Mitzenmacher [13] also notes that it is difficult to distinguish power law and lognormal distributions. We do not attempt to make such a distinction because the property of having a small number of bidders being responsible for a disproportionate amount of bidding activity is shared between the two distributions, and that is the relevant observation for our analysis.

follow a power law for both markets; graphs are shown in Figure 2.2. The relatively low range in the number of auctions won in the Canon market (1 to 22) may be responsible for the slightly poorer fit.

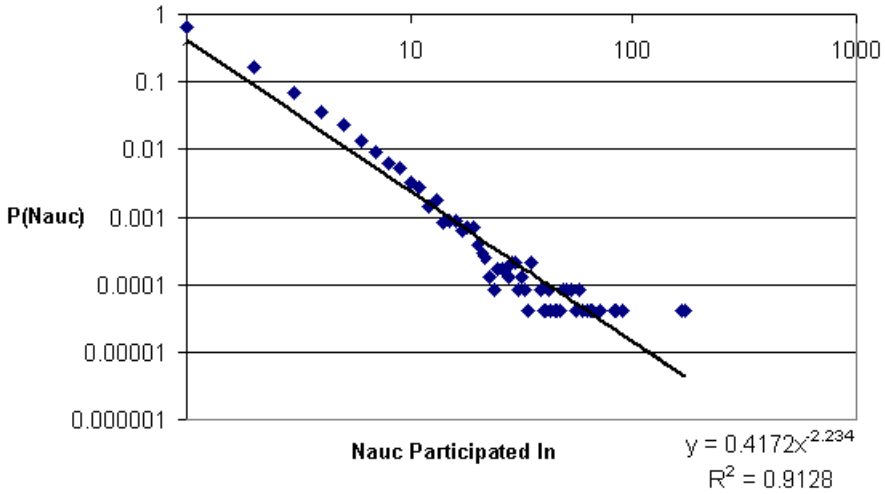
These data suggest that most bidders in the market were only interested in winning one item. As further support for this claim, winning bidders tend not to participate in other auctions after winning an item. In the Canon market, 2422 of the 3638 winners who one only one item (67%) participated in only one auction (the auction that they won). In and of itself, this statistic is not conclusive, since from the above section we see that 66% of all bidders (non-winners and winners alike) participated in only one auction.

More revealingly, when we examine the 1216 single auction winners who did participate in more than one auction (but won only one), 1070 of the wins (88%) came in the last auction in which the bidders participated. In other words, even though these winners participated in more than one auction, it appears that they were only interested in winning one, since they stopped bidding in other auctions after their win. If winning an auction had no effect on continued participation in the market, then one would expect the above percentage to be closer to 50%. Taken together, 2422+1070=3492 of the 3821 unique bidders (91%) in the Canon market won only one auction and did not participate in any auctions after winning.

The data for the LCD market are similar. In this market, 4332 of the 6708 bidders who won only one auction (65%) participated in only one auction. Of the 2376 single-item-winners who participated in more than one auction, 2039 (86%) did not participate in any auctions after their win. Taken together, 4332+2039=6371 of the 7227 unique bidders (88%) in the LCD market won only one auction and did not participate in any auctions after winning.

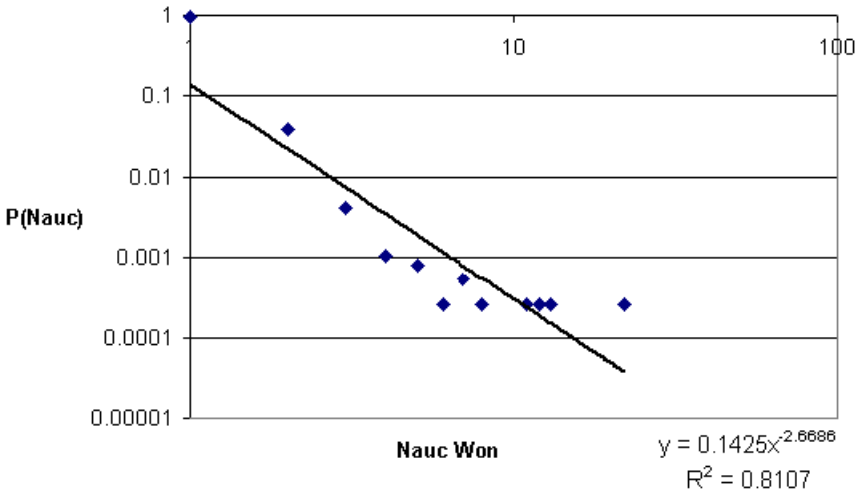


(a) Canon

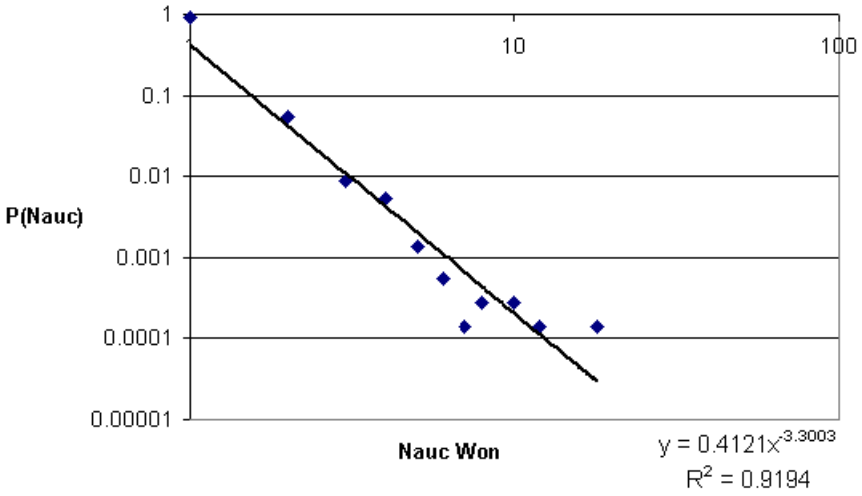


(b) LCD

Figure 2.1: Distribution of Number of Auctions Participated In



(a) Canon



(b) LCD

Figure 2.2: Distribution of Number of Auctions Won

Chapter 3

The Substitutes Problem and Our Solution

In this chapter, we provide a more specific definition of the substitutes problem and motivate the approach we adopt to solve it. We then detail the methods for each of the three parts of our solution. In the next chapter, we discuss the results of applying our solution to the data sets.

3.1 Problem Definition and Motivation

Informally, two goods can be defined as *substitute goods* if obtaining one reduces or eliminates demand for obtaining the other. Lehmann et al. [9] provide a more formal definition where they define two items as *gross substitutes* iff the demand for one of the items does not decrease when the price of the other item increases.[1] In contrast, if the price of one of a pair of *complementary goods* goes up, the demand for the other complementary good may go down.

An example of a pair of substitute goods might be two similar models of digital camera. If the price of one of the digital cameras goes up, there is no reason for demand for the other camera to go down. An example of a pair of complementary goods might be a digital camera and a memory card. In this case, a buyer might gain added value from having both, since a camera is less useful without a memory card. If the price of the memory card goes

[1] Lehmann et al.'s definition of gross substitutes is more general than the one we use in this paper. In particular, they define gross substitutes for arbitrarily large sets of goods and in cases where individuals might want to buy multiple items from a set.

up, the buyer's demand for the camera might well go down. Similarly, if the price of the camera goes up, the buyer's demand for the memory card might go down.

The substitutes problem that we address in this paper can phrased as follows: given a segment of the online auction market, how does one automatically determine the communities of substitutable goods?[2] We propose a three-step approach to solving this problem. We first generate a network with auctions as nodes and edges drawn between auctions with shared bidders; next, we apply a community detection algorithm to the largest MCC of the network; finally, we characterize the communities found by applying a keyword extraction algorithm.

This is not the only approach that could be used. For example, an alternate approach would be to apply natural language processing and document similarity algorithms directly to auction listing text. However, using an auction-auction network generated from bidder data allows us to leverage the "revealed preference" information that bidders implicitly express when they interact with the site. If the assumption that bidders will generally limit their auction activity to auctions with substitutable goods is valid, then one would expect our approach to produce good results.[3] The results of Reichardt and Bornholdt [20] suggest that this assumption is fair. The fact that the vast majority (over 90%) of winning bidders win only one item further supports the idea that, at least in the two markets we examine, bidder behavior is a strong indicator of substitutability.

Leveraging knowledge implicit in networks has been used effectively in other areas, such as web search. For example, the Google search engine considers the importance of pages that link into a given page, in addition to document similarity measures, when deciding page relevance [17].

In the next three sections, we discuss methods for each of the three parts of our solution.

[2]Strictly speaking, we address a closely related problem, that of identifying communities of identical (or very similar) goods. Not all identical goods communities are also "pure" substitutes communities because there are cases where complementary relationships can exist even within identical goods—for example, some buyers might gain added utility from having multiple copies of a certain type of memory cards. However, other buyers might not want to have multiple copies, so for these buyers, the items in that category of memory card would indeed be substitutes. If we wanted to get a closer approximation to substitutes communities, we would exclude from our data set any bidders who win multiple auctions (similar to winning-bidder filtering, which we discuss in Section 3.2.2). In practice, there is considerable overlap between identical goods communities and substitutes communities, and both are applicable to the larger motivating problem of item categorization.

[3]We also assume that bidders will use the same user name for their transactions.

3.2 Generating the Network

In this section, we give an overview of various ways of constructing an auction network to solve the substitutes problem. Generating the network is the first step in our solution.

The basic undirected auction network that we propose is constructed with auctions as nodes. In the most general case, an edge is drawn between any two auctions that share a common bidder. More formally, we define our auction-auction graph $G = (V, E)$ where $v \in V$ iff v is an auction in the market and $e = (v_1, v_2) \in E$ iff $\exists$ a bidder b s.t. b is a bidder in both auctions v_1 and v_2. An alternate graph representation could be a bipartite graph with both bidders and auctions as nodes. However, since we are only interested in classifying auctions, collapsing the bipartite graph into an auction-only graph is a reasonable first step.

3.2.1 Edge Weighting

We consider both weighted and unweighted versions of the network. We propose a natural way of assign weights to edges—weighting edges by the number of shared bidders between any two auctions. This method makes sense because a greater number of shared bidders should indicate a greater likelihood of substitutability.

3.2.2 Edge Filtering

One potentially undesirable property of the way we generate edges is that a small fraction of bidders can be responsible for a disproportionate number of edges, due to the fact that the degree distribution of bidder-auction edges follows a power law. A bidder that participates in n auctions generates $n(n-1)/2$ edges in our network. Thus, the bidder in the LCD data set who participated in 171 distinct auctions was responsible for generating 14535 edges—more than 10% of the 144822 edges in the network—even though he accounts for less than 0.02% of the bidder population.[4] We do not want the community structure in our network to be dominated by any one bidder unless we have reason to believe that that bidder provides more meaningful information than other bidders. The dominance of "power bidders" would be especially problematic if such bidders do not provide correspondingly powerful information about substitutability in their bidding patterns—for example, if they bid more or less randomly, without regard to substitutability. Unfortunately, the bids of the bidder who participated in 171 auctions spanned items of varying brand, screen size, and

[4]It is possible that other users also generated some of the same edges. However, the fact remains that a highly active bidder can generate a disproportionate number of edges.

model. Many of these bids were extremely low relative to the ending price of the auction, and he was not the high bidder in any of the 171 auctions that he bid in.

To address the problem of "meaningless" edges adding noise to our network, we propose two ways to filter the edges generated:

1. Price threshold: When generating edges, only consider bids that are at least a certain fraction f of the ending price. More formally, let b_v denote the maximum value of a bid placed by bidder b in auction v and $maxbid(v)$ denote the maximum bid recorded in that auction. In our graph, we then define edges $e = (v_1, v_2) \in E$ iff $\exists$ a bidder b s.t. b is a bidder in both auctions v_1 and v_2 and both $\frac{b_{v_1}}{maxbid(v_1)} >= f$ and $\frac{b_{v_2}}{maxbid(v_2)} >= f$. This filter is aimed at removing "non serious" lowball bids that might be less category selective. We take a fraction of the ending price rather than an absolute cutoff because we have no easy way of determining what the true value of any auctioned item is.

2. Winning bidders: When generating edges, only consider bids placed by bidders who won exactly one auction. We define edges $e = (v_1, v_2) \in E$ iff $\exists$ a bidder b s.t. b is a bidder in both auctions v_1 and v_2 and the number of auctions that b won is equal to 1. Using similar logic to that in the price thresholding case, if the bidder wins an auction, it might be more likely that the bidder is serious and directed in bidding. Furthermore, the fact that the bidder only won one auction might be indicative of a more substitute-focused approach to bidding. However, this filter might be too restrictive—that is, one would be filtering out valuable information provided by serious bidders who simply failed to win an auction.

As an illustration of how edge filtering methods can reveal community structure, consider Figures 3.1, 3.2, and 3.3, where we show spring-model energy-minimization representations of various edge filters applied to the same starting network. As we filter edges, natural communities of more closely connected auctions emerge. These figures should not be interpreted too literally, however, since a community detection algorithm could still reveal structure that is not readily apparent in the graphical representation. We discuss community detection algorithms in the next section and more formally assess the effect of edge filtering in the next chapter.

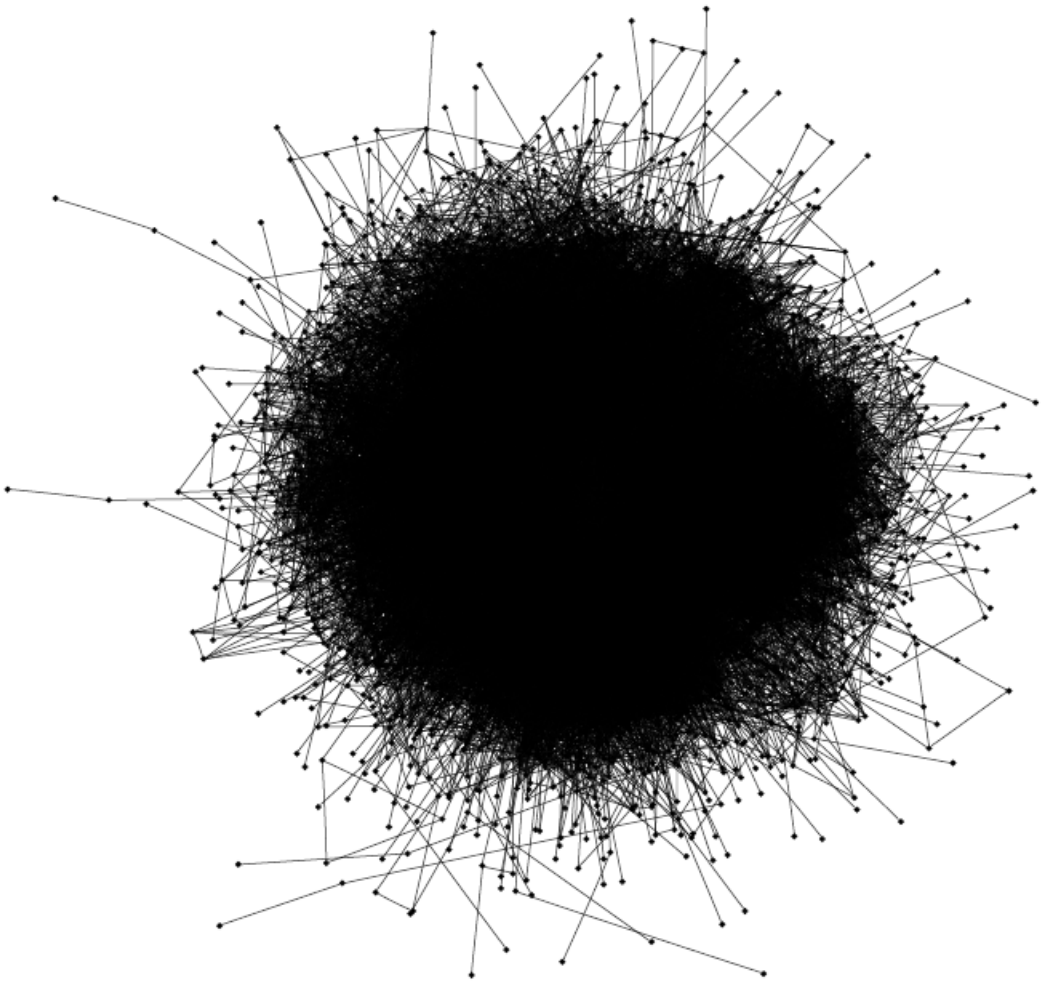


Figure 3.1: The Largest MCC of the Unweighted Canon Network with No Edge Filtering. The graph was plotted using the neato utility of Graphviz, which implements a spring-model energy-minimization algorithm for graph visualization. The nodes in the graph are auctions and edges are drawn between any two auctions that share a bidder.

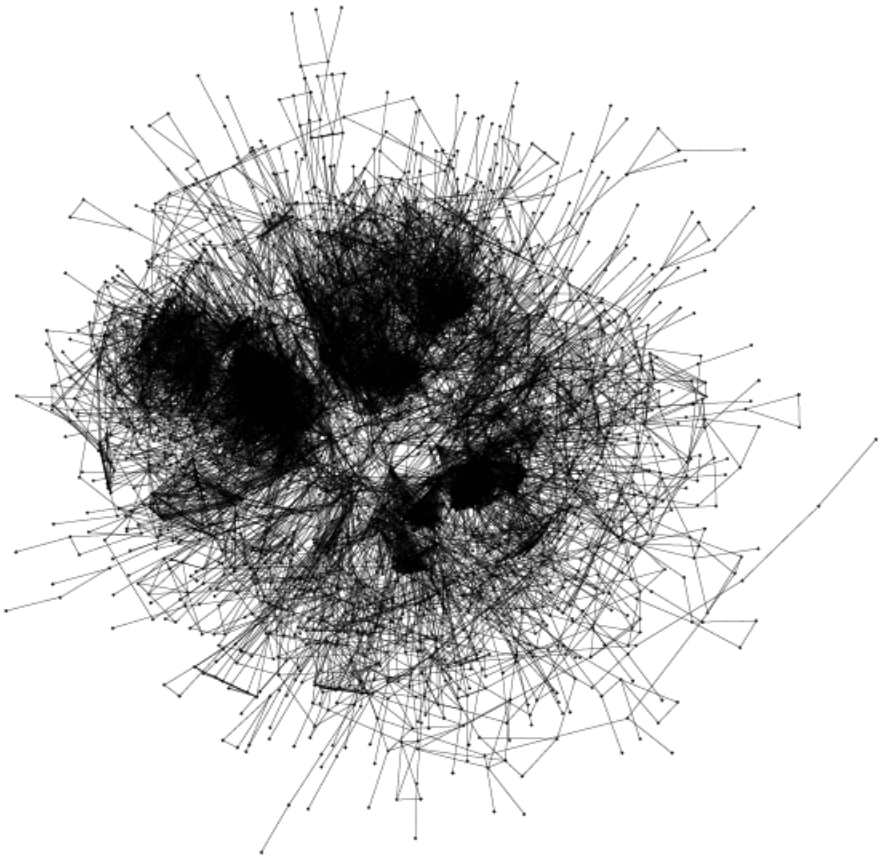


Figure 3.2: The Largest MCC of the Unweighted Canon Network with Price Threshold 0.8 Edge Filtering. More structure is visible in this network than in the unfiltered network (Figure 3.1).

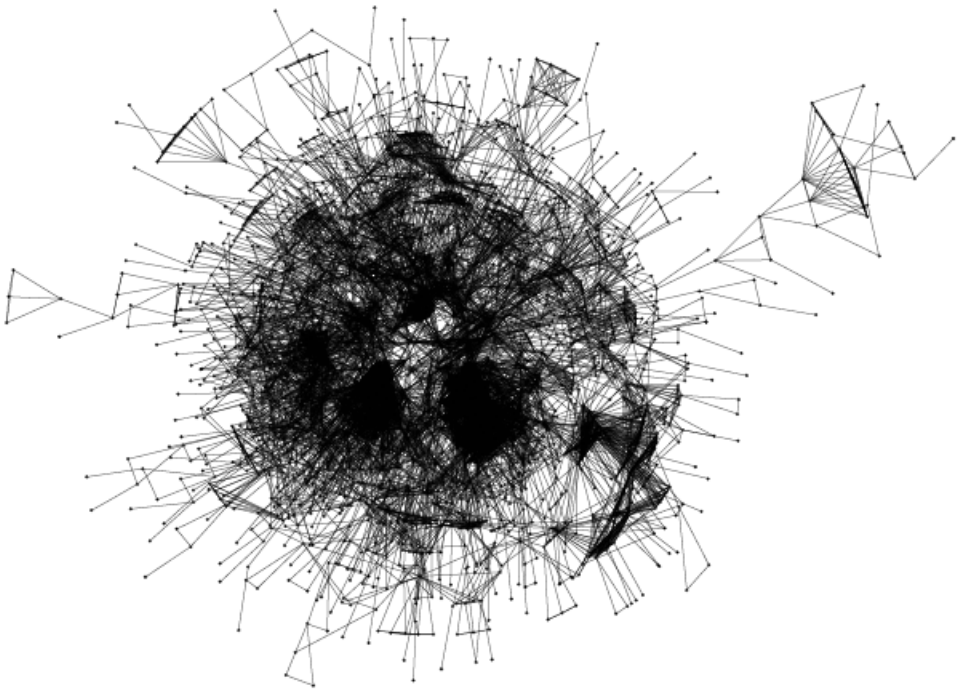


Figure 3.3: The Largest MCC of the Unweighted Canon Network with Winning Bidders Edge Filtering. More structure is visible in this network than in the unfiltered network (Figure 3.1).

3.3 Applying Community Detection

Having generated the network, the next step in our solution involves identifying communities within it. In this section, we begin by defining communities and ways to evaluate a community assignment for a network. We then discuss the community detection algorithm that we use.

3.3.1 What is a Community?

Community detection in networks is a growing area of research because it can help to reveal underlying structure [18]. For example, Flake et al. [5] found that web pages tend to cluster into communities of semantically similar pages; community structure has also been examined in social and biological networks [6]. Intuitively, a community is defined as a subset of nodes that are connected more strongly to each other than to the rest of the network. For example, a social network of all college students in the United States where edges are defined by friendships might naturally have communities defined along college lines. Reichardt and Bornholdt [19] propose one formal definition of a community. Given a graph G with N nodes and M edges, a community of n nodes and m edges is one satisfying:

$$\frac{2m}{n(n-1)} > \frac{2M}{N(N-1)} > \frac{m_{nN}}{n(N-n)} \tag{3.1}$$

where m_nN is the number of edges connecting the community to the rest of the network. Each of these terms represents an edge density—the number of edges divided by the maximum number of edges. The first inequality requires that within-community density be greater than the average network density, while the second inequality requires that the average network density be greater than the density of edges leaving the community.

For a given graph $G = (V, E)$, the community detection problem can be formalized as a partitioning problem subject to a constraint. For all $v \in V$ we need to assign a $c_v \in 1, 2, ..., n_c$ where n_c is the number of communities. We want this partitioning to match our intuition as to what a good community is—that is, the partitions should satisfy some definition of community, such as the one found in Equation 3.1, and/or maximize some metric that assesses a proposed community partitioning.

The community detection problem is difficult because there may be multiple ways to divide any graph into acceptable communities, and furthermore, the number of "optimal" communities is often not known beforehand. Modularity is a global metric that has been widely used to compare different community divisions and determine an "optimal" one

[16]. Let e_{ij} be the fraction of all edges in the network that lie between community i and community j and d_v be the degree of vertex v. Then the modularity Q of a network and community division is:

$$Q = \sum_i e_{ii} - {a_i}^2 \tag{3.2}$$

where:

$$a_i = \frac{1}{2M} * \sum_{v \in i} d_v \tag{3.3}$$

a_i gives the fraction of ends of edges in the network that are in the community i. ${a_i}^2$ then represents the expected value of the fraction of edges in the network that would fall wholly in community i if the edges had been assigned by chance, but keeping the same communities (that is, edges with both ends in i).

Q thus represents the fraction of edges in the network that fall within communities (given by e_{ii} in equation 3.2) less the expected value of that fraction if the edges had been assigned by chance, but keeping the same communities (given by ${a_i}^2$).

A value of $Q = 0$ indicates no community structure, while a value of Q approaching the maximum of 1 represents the presence of strong community structure.[5] Newman [16] found that real world networks with strong community structure tend to have Q values of at least 0.3. The highest known value of Q in the literature for an unweighted real-world network is $Q = 0.75$, which came from a reviewer network from Amazon.com [2].

Modularity generalizes naturally to graphs with weighted edges [15]. In this case, we define M as the sum of all the weights of edges in the network, e_{ij} as the sum of all the weights of edges between communities i and j divided by M, and d_v as the sum of all the weights of edges touching vertex v.

3.3.2 Finding Communities

Armed with this metric, we are now prepared to find communities in a network.[6] Community detection algorithms generally assume that the network is connected, and consistent

[5] In practice, it is possible to generate networks and community assignments where $Q < 0$. Furthermore, for any given network, it may be impossible to attain $Q = 1$. In earlier work, Newman [14] normalized the value of Q; however, he argues that the unnormalized Q is more informative [16].

[6] Other metrics and methods to detect communities exist. For a review, see Danon et al. [3]. We chose to use a modularity based approach because it is widely used [3] and has a relatively fast runtime, which we discuss later in this section.

with previous work [2], we take the largest maximally connected component (MCC) of the network for analysis. The goal, then, is straightforward—we simply wish to find a community division that maximizes Q for the MCC of a network. Unfortunately, finding the maximum Q for a graph is at least exponentially hard in the number of nodes n if one were to search over all possible community divisions. As Newman et al. [16] note, the number of ways of partitioning a set of n nodes into c nonempty sets is given by the Stirling set number $S(n, c)$. To exhaustively search the entire space of community partitions, one would need to consider $\sum_{c=1}^{n} S(n, c)$ partitions. Since $S(n, 2) = 2^{n-1} - 1$, the search would be at least exponentially hard.

Since an exact solution is intractable for most large data sets, Newman et al. [16] propose a greedy forward-selection method ("Greedy Q") where one starts with each node in its own community and iteratively joins communities based on the greatest increase in Q. This method has been demonstrated to reliably identify communities in both artificial networks and real-world networks [2].

Place every node in its own community;
$N_{Communities}$ = number of nodes;
while $N_{Communities} > 1$ **do**
  $MaxQ = -\infty$;
  for *each pair of distinct communities* i *and* j **do**
    if Q *from joining* i *and* $j > maxQ$ **then**
      $MaxQ = Q$;
    end
  end
  Join the i and j that gave $MaxQ$;
  $N_{Communities} = N_{Communities} - 1$;
end
Return the community assignment corresponding to $MaxQ$;

Algorithm 1: Greedy Q

For this thesis, we implemented the greedy algorithm as optimized in Clauset et al. [2] ("Optimized Greedy Q"). Rather than calculating Q from scratch for each potential community join, the algorithm keeps an array ΔQ, where entry i,j represents the change in Q that would result from joining communities i and j. The insight behind the optimization is that ΔQ can be initialized easily and that only a few entries in ΔQ need to be changed

each time two communities are joined.

Place every node in its own community;
Initialize a;
Initialize ΔQ;
$N_{Communities}$ = number of nodes;
while $N_{Communities} > 1$ **do**
  Join the i and j into community j corresponding to the max value in ΔQ_{ij};
  Update ΔQ;
  Update a;
  $N_{Communities} = N_{Communities} - 1$;
end
Return the community assignment corresponding to $MaxQ$;

Algorithm 2: Optimized Greedy Q

Four of the lines in the the pseudocode for Optimized Greedy Q require elaboration: the two initialization steps, and the two update steps. A derivation of these steps is found in Appendix A.

1. Initialization of a. This initialization is quite straightforward. Since each community consists of one node,
$$a_i = \frac{d_i}{2M} \tag{3.4}$$

2. Initialization of ΔQ. There are two different initial values of ΔQ_{ij} depending on whether i and j are connected.[7] If i and j are connected,
$$\Delta Q_{ij} = \frac{1}{m} - \frac{d_i * d_j}{2m^2} \tag{3.5}$$
If i and j are not connected,
$$\Delta Q_{ij} = 0 \tag{3.6}$$

3. Updating ΔQ. As noted before, only a few elements of the ΔQ matrix need to be updated after community i is joined to community j to form community j'. In particular, we need to update elements in the jth row and column (and also delete the ith row and column). Furthermore, we do not need to update every item in the jth

[7]The value in Equation 3.5 is twice as much as the corresponding equation 8 in Clauset et al. [2]. The discrepancy is due to a typographical error in their paper.

row and column–we only need to update the items that correspond to a community k that was connected to at least one of i and j prior to their join.
If k is connected to both i and j:

$$\Delta Q'_{jk} = \Delta Q'_{kj} = \Delta Q_{ik} + \Delta Q_{jk} \tag{3.7}$$

If k is connected to i but not j:

$$\Delta Q'_{jk} = \Delta Q'_{kj} = \Delta Q_{ik} + 2a_j a_k \tag{3.8}$$

If k is connected to j but not i:

$$\Delta Q'_{jk} = \Delta Q'_{kj} = \Delta Q_{jk} + 2a_i a_k \tag{3.9}$$

4. Updating a. This update is quite simple.

$$a'_j = a_j + a_i \tag{3.10}$$

ΔQ is stored in two ways. Rows are stored as both balanced binary trees and as max-heaps. In addition, a separate max-heap H stores the maximums of each row in ΔQ. Each update step takes $O((|i| + |j|) \log n)$ time. We make $|i| + |j|$ insertions of cost $O(\log n)$ into the balanced binary trees. We also make $|i| + |j|$ reheaps for the row max-heaps and up to $|i| + |j|$ reheaps for H, each with cost bounded by $O(\log n)$. There are at n update steps, and $|i| + |j|$ is bounded by n as well, so the runtime of the algorithm is $O(n^2 \log n)$.[8]

An example illustrating the results of applying community detection to a network is shown in Figure 3.4.

[8] $|i| + |j|$ is in fact often much less than n and, by making additional assumptions, one can develop a tighter bound [2].

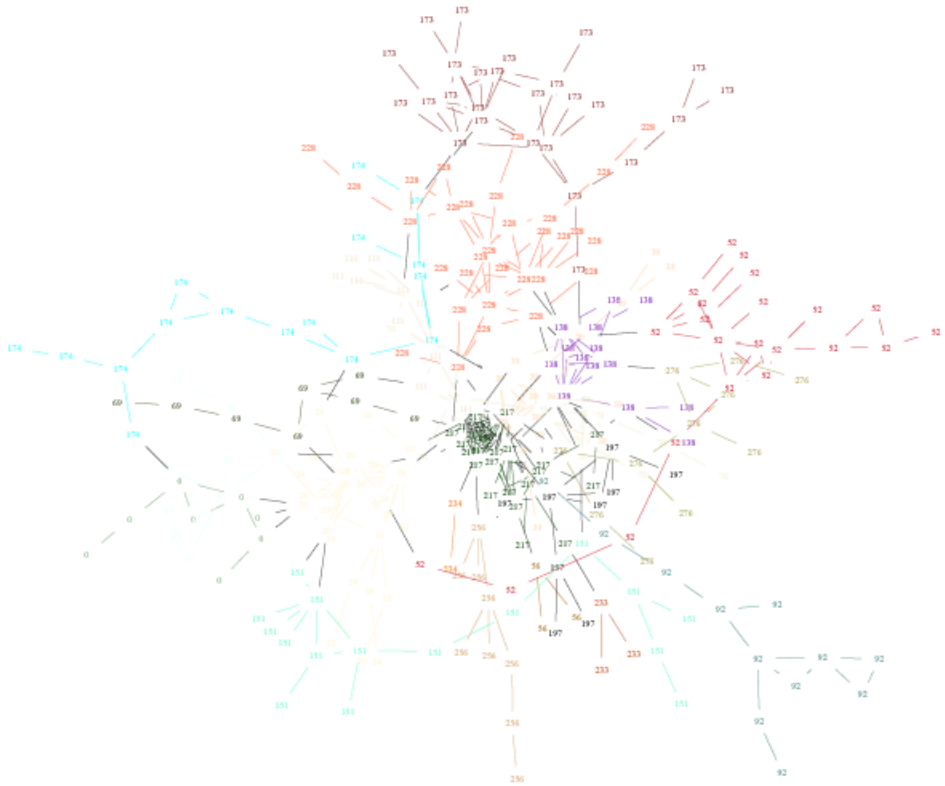


Figure 3.4: Community Assignments for a Sample Network. In this network, nodes assigned to different communities are given different colors and number labels. The network was derived from a 10% sample of the auctions in the LCD market. Nodes are auctions and edges are drawn between any two auctions that share a common bidder. The network was plotted using neato. We depict the 10% sample rather than the full network because the visualization tool performs better on smaller networks. It is interesting to note that auctions which cluster together in the spring-model energy-minimization visualization of the graph tend to be assigned to the same community.

3.4 Characterizing Communities via Keyword Extraction

The community partitioning of a network is only useful for the substitutes problem if we are able to determine what type of good each community represents. In this section, we present a simple algorithm for automatically extracting uniquely representative descriptive keywords from the auction titles in a community. There are other well-established methods for keyword extraction, such as term frequency-inverse document frequency (tf-idf) [23]. One advantage to the approach we use is that it allows for potentially better "tuning" because we can threshold by two independent parameters.

The algorithm is based upon the intuition that a representative keyword for a community should satisfy the following two properties:

1. Widely shared: the keyword should be shared by a large percentage of the auctions in the community.

2. Overrepresented: the percentage of auctions having the keyword within the community should be greater than the percentage of auctions in the entire network having the keyword in a statistically significant manner.

The motivation for property (1) is obvious—a keyword can only be descriptive of a community if a large fraction of its members share it. Property (2) is required so that we can highlight the keywords that uniquely define a community. For example, in the LCD market, almost all of the auctions in any given community will have LCD in the title, but we would not want to consider it a unique descriptive keyword for a particular community.

More formally, in order for a keyword to be considered a descriptive keyword for a community:

1. The keyword must be shared by at least a fraction p_c of the auctions in the community.[9]

2. The fraction p_c must be statistically significant relative to the population proportion p_g at a confidence level a (single-tailed Z-test). We define $z = (p_c - p_g)/s$, where p_g is the fraction of auctions in the global set of all n auctions that share the keyword and

[9]We define an auction as "sharing" a keyword if the title of the auction contains that keyword. There are many possible refinements to this method, such as examining auction page text in addition to title text. However, using only titles is a reasonable starting point, and, as we shall see, works well in practice. Sellers also have incentive to make auction title text descriptive summaries of their items, since when bidders browse listings, only the auction title is visible. Keywords were space-delimited.

$s = \sqrt{p_g * (1 - p_g)/n}$ is the standard deviation. We then define $a = \mu prob(z)$ where $\mu prob(z)$ is the upper probability of the μ distribution.[10]

Setting the parameters p_c and a determines how strongly one wishes to enforce these two properties. A higher p_c and a lower a would impose more stringent requirements on the keywords extracted.

[10]In practice, the Z-test requires a sample size of at least 30 to be significant. Most of our major communities are at least size 30, so this requirement is not an issue.

Chapter 4

Substitutes Problem Results

In this chapter, we present the results of applying our solution to the substitutes problem to the eBay data. The key findings are that (1) edge filtering contributes to significantly better community structure, (2) edge weighting also contributes to slightly better community structure, and (3) our solution to the substitutes problem using price thresholding and edge weighting produces very reasonable categories of substitutable goods.

We demonstrate the first result both quantitatively and qualitatively. First, we show that edge filtering increases modularity—indeed, the edge filtered networks have modularity scores similar to the highest recorded modularity for real-world networks. Second, we show that edge filtering, when combined with keyword extraction, results in communities that have qualitatively better keywords. We demonstrate the second and third results through a similar qualitative discussion of the keywords extracted from the communities found.

We conclude this section by discussing the significance of the results and some limitations to our method.

4.1 Edge Filtering and Modularity

In this section, we present data demonstrating that both methods of edge filtering that we propose—price thresholding and single winner filtering—increase modularity with respect to both unfiltered networks and to "control" networks that have the same number of edges as our filtered networks, but with edges randomly filtered. In the next section, we show that this increased modularity corresponds to qualitatively better communities, suggesting that edge filtering reveals otherwise-hidden real community structure in each market.

4.1.1 Price Thresholding

As discussed earlier, price-threshold edge filtering consists of only including edges generated from bids that were at least some fraction f of the ending price of the auction. We begin this subsection by discussing some basic properties of the edge filtering method and guidelines for choosing a fraction f. We then discuss the effect of thresholding on modularity.

We generated networks with varying levels of price thresholding. Descriptive statistics for these networks are found in Tables 4.1 and 4.2. In a picking a threshold f, one would want to select a fraction f that filters a significant number of edges while leaving the size of the largest maximally connected component (MCC) relatively unchanged. It is important to keep as many nodes as possible, since ultimately the communities and categories will come from these nodes, and if the largest MCC is not a significant fraction of the overall market, then potential categories of goods in the market might be lost. At the same time, it is important to filter out as many uninformative edges as possible.

Threshold	Edges	Edges in Largest MCC	Nodes in Largest MCC
Unfiltered ($f = 0$)	43737	43690	3173
Filtered $f = 0.2$	36505	36447	3141
0.4	30535	30471	3099
0.6	22830	22752	3012
0.8	13035	12858	2710
0.9	7452	6930	2103
1	926	231	22

Table 4.1: Canon Network Statistics with Price Threshold Filtering

Threshold	Edges	Edges in Largest MCC	Nodes in Largest MCC
Unfiltered ($f = 0$)	144822	144503	6024
Filtered $f = 0.2$	113422	113060	5934
0.4	92055	91660	5813
0.6	60955	60502	5621
0.8	28918	28206	5072
0.9	15079	14009	4171
1	1726	153	18

Table 4.2: LCD Network Statistics with Price Threshold Filtering

One desirable property of price thresholding in the networks we examine that quickly becomes apparent from these statistics is that we can remove many edges (via a high threshold) before we start losing a significant number of nodes in the MCC. For example, in the Canon data set at threshold 0.6, 48% of the edges are filtered at a "cost" of only 5% of the

nodes. To illustrate the tradeoff between edges and nodes, we plot, in Figures 4.1 and 4.2, the percent of edges and the percent of nodes in the MCC as a function of our threshold fraction f.

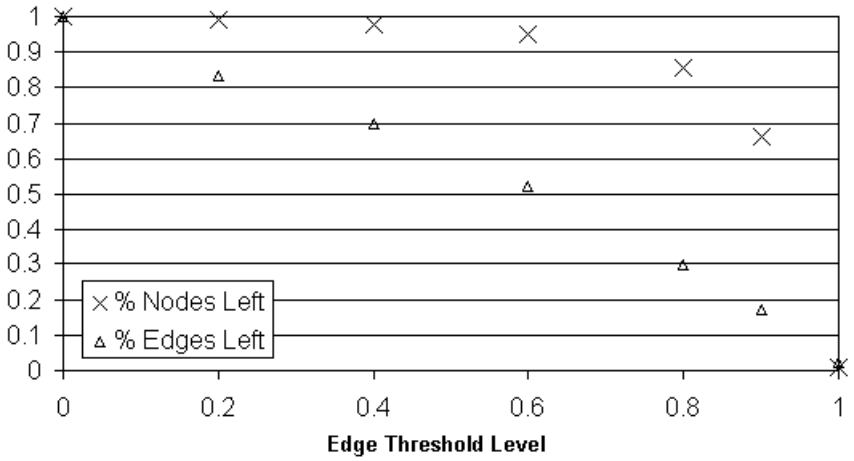


Figure 4.1: Canon Edges and Nodes vs. Price Threshold

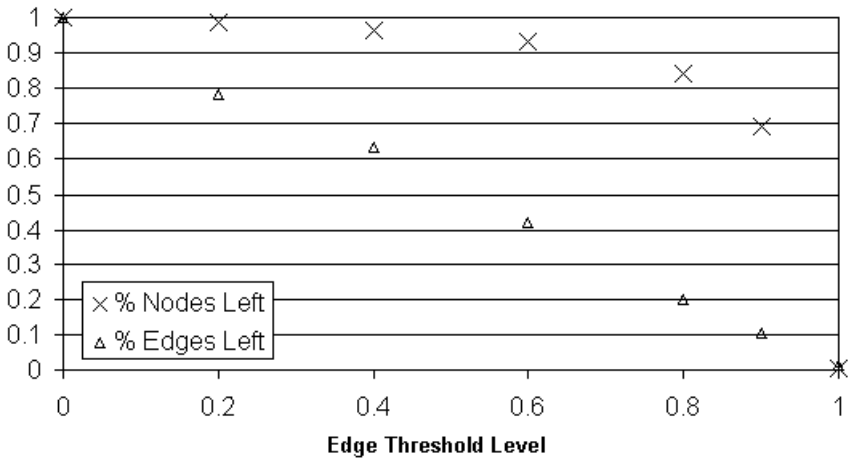


Figure 4.2: LCD Edges and Nodes vs. Price Threshold

In order to assess whether filtering edges by price threshold increases modularity, we ran the community detection algorithm for the networks at threshold 0 (unfiltered), 0.6, and 0.8. We generated unweighted networks because we wanted to compare our modularity scores with those in the existing literature, which have been generated using unweighted

networks.

We found that price thresholding increased the modularity. In the Canon market, the base modularity at filtering threshold 0 (unfiltered) was 0.51. Filtering at threshold 0.6 increased the modularity to 0.66, and filtering at threshold 0.8 further increased the modularity to 0.77. In the LCD market, the base modularity was also 0.51. Filtering at threshold 0.6 increased the modularity to 0.63, and filtering at threshold 0.8 further increased the modularity to 0.72. These data are summarized in Table 4.3.

As noted earlier, real-world networks with community structure tend to have Q scores of 0.3 and above; thus, even the unfiltered eBay networks we construct show evidence of community structure. However, it is significant that filtering increased the modularity score. The highest recorded value of modularity for a real-world network in the literature was 0.75, and the modularity of the Canon network at threshold 0.8, 0.77, was higher.

To further establish the effectiveness of edge filtering, we generated "control" filtered networks that filtered the same number of edges as the price thresholded networks, but with edges filtered randomly. For example, the control network for the Canon threshold 0.8 network was a network with 13035 edges (the same number of edges as the Canon threshold 0.8 network) constructed by randomly removing 30702 edges from the base network, which had 43737 edges. One caveat is that while the full control network has the same number of edges as its matched edge filtered network, the largest MCC of the control network has a slightly different number of edges and nodes from the largest MCC of the edge filtered network. In all the price thresholding cases, this difference was less than 5% of the number of edges in the filtered largest MCC. The network statistics for the price-threshold control networks can be found in Tables B.1 and B.2 in Appendix B.

We ran the community detection algorithm for these networks and compared the Q scores. The results are shown in the tables below. While removing random edges appears to increase Q relative to baseline, in every case the control network had a lower Q score than the paired price thresholded network. This result demonstrates that the greatly increased modularity in the price threshold networks is not simply an artifact of removing edges—rather, it is the result of removing edges intelligently, thus revealing real community

structure.[1]

Canon Network	Q from Price Thresholding	Q of Control
Unfiltered ($f = 0$)	0.51	0.51
Filtered $f = 0.6$	0.66	0.58
0.8	0.77	0.62
LCD Network	**Q from Price Thresholding**	**Q of Control**
Unfiltered ($f = 0$)	0.51	0.51
Filtered $f = 0.6$	0.63	0.53
0.8	0.72	0.56

Table 4.3: Canon and LCD Network Modularity with Price Threshold Filtering

In interpreting these results, it is important to keep in mind that the greedy community detection algorithm does not guarantee the optimal solution. There is also no guarantee that given two networks N_1 and N_2 where the optimal Q for N_1 is greater than the optimal Q for N_2, that the greedy algorithm will arrive at a higher Q for network N_1 than for network N_2. Thus, it is possible that the trends observed above are somehow due to an inherent bias in the algorithm. Nonetheless, the fact that the trends are consistent in all the cases examined makes this possibility less likely.

4.1.2 Winning Bidders

In addition to examining price filtering, we also examined winning bidder filtering—only considering bids from bidders who won exactly one auction. Basic descriptive statistics along with Q scores for these networks are contained in Tables 4.4 and 4.5. As before, we compared the Q scores to a control network with the same number of edges. In this case, the control networks had similar numbers of edges as their corresponding winning bidder filtered networks but had over 20% more nodes. Thus, the comparison to control networks in this case might be less informative. The network statistics for the winning-bidders control networks can be found in Tables B.3 and B.4 in Appendix B.

[1]Increased modularity is not necessarily good in and of itself—presumably, there is some natural level of community structure inherent to each market that we examine, and one could imagine that some methods of edge filtering might bias the network by "artificially" increasing modularity, resulting in community assignments of high modularity but low semantic value. For example, given a fully connected market (where there is no true community structure), one could apply a biased edge-filtering method that selectively removes edges in order to form "artificial" communities in the filtered network. However, our methods of filtering edges are not biased in so obvious a manner, and in Section 4.2, we show that the increased modularity of our price-threshold-edge-filtered networks correlates with communities that have qualitatively better keywords. Thus, the increased modularity in this case corresponds to the discovery of real community structure.

Winning bidder filtering, like price thresholding, seems to increase modularity relative to baseline and control. In the Canon network, the Q of the winning bidder filtered network was 0.71, while the Q of the unfiltered network was 0.51, and the Q of the control network was 0.61. In the LCD network, the Q of the winning bidder filtered network was 0.62, while the Q of the unfiltered network was 0.51, and the Q of the control network was 0.53. These results are shown in Table 4.6.

However, winning bidder filtering has a few disadvantages when compared to price thresholding. First, it seems like this method of filtering is too strict, since we lose more nodes in the largest MCC for a similar number of edges removed relative to price thresholding. As noted before, we want to keep the largest MCC largely intact because ultimately our community categories will come from the largest MCC. For example, price thresholding at 0.6 for the LCD market has 60955 edges and a largest MCC of 5621 auctions, while winning bidder filtering has almost the same number of edges (59136) but a largest MCC of only 4281 auctions. Second, this method is less tunable, since we cannot vary the selectivity of the winning bidders filter to get a desired number of nodes in the largest MCC.

Network	Edges	Edges in Largest MCC	Nodes in Largest MCC
Unfiltered	43737	43690	3172
WBF	12142	11931	2178

Table 4.4: Canon Network Statistics with Winning Bidder Filtering (WBF)

Network	Edges	Edges in Largest MCC	Nodes in Largest MCC
Unfiltered	144822	144503	6024
WBF	59136	58715	4281

Table 4.5: LCD Network Statistics with Winning Bidder Filtering (WBF)

Canon Network	Q	Q of Control
Unfiltered	0.51	0.51
WBF	0.71	0.61

LCD Network	Q	Q of Control
Unfiltered	0.51	0.51
WBF	0.62	0.53

Table 4.6: Canon and LCD Network Modularity with Winning Bidder Filtering (WBF)

4.2 Edge Filtering and Community Keywords

In this section, we demonstrate that in addition to improving modularity, edge filtering also results in communities with qualitatively better keywords. Ultimately, the keywords are the most informative output of our method, since they enable us to define the types of goods

(hopefully substitutes) that are contained in each community of auctions that we find. We examine price-threshold edge filtering, since, as discussed in the previous section, we believe that it is a more promising method that winning-bidders filtering. Additionally, we focus our analysis in this section on the Canon market. A qualitative analysis of the LCD market gave similar results and can be found in Appendix C.

We compared the results of applying our solution to the unweighted Canon network with price thresholding at $f = 0$ (unfiltered) to the results of applying it to the unweighted Canon network with price thresholding at $f = 0.8$. We evaluated the communities and keywords found using qualitative standards for within-community and across-community keywords, as discussed in the next paragraph.

Qualitatively, within a given community, a good set of keywords should be one that defines a category of substitutable goods. In the Canon market, we might want keywords to correspond to specific models of cameras. Other possible good keywords would be those specifying key attributes like camera resolution. We would not want too many keywords for a community—it would be surprising if many different camera models were grouped together in the same community, since different models generally make poor substitutes. Across communities, we would want minimal overlap between keywords, since each community should define a unique set of substitutable goods. In addition, we would want enough communities to be able to identify a range of distinct substitutes communities.

For the network with $f = 0$, there were 22 communities ranging in size from 2 to 1256. We define a major community as one whose size is at least 1% of the total number of auctions in the MCC. In the network with $f = 0$, there were 4 such communities. We applied keyword extraction following the methods outlined in Section 3.4 with $a = 0.01$ and $p_c = 0.3$[2] and examined the keywords for those 4 communities. The largest community did not have any significant keywords.[3] The remaining 3 communities had 4 to 5 significant keywords each. The major communities with significant keywords are depicted in Figure 4.3. One community had no specific models of camera as a keyword, 1 community had 1 specific model (s2), and 1 had 2 specific models (a410 and a520). It is questionable whether the a410 and a520 cameras are good substitutes, since they vary on many dimensions, including resolution (3.2 vs 4.0 megapixels). Across communities, there was no overlap between camera model keywords.

[2] $a = 0.01$ requires that the keyword must be overrepresented at a 99% significance level, and $p_c = 0.3$ requires that the keyword must be shared by at least 30% of the auctions in the community. We tried to find parameters to give good results for the $f = 0$ unweighted network and kept the same parameters for all of our analyses.

[3] The first keyword satisfying $p_c = 0.3$, for any level of a, is the not-too-informative "digital" with $a = 0.07$.

For the network with $f = 0.8$, there were 26 communities ranging in size from 3 to 637. Of these, 11 were major communities. We applied keyword extraction with the same parameters and examined the keywords. Nine of the major communities had significant keywords, with a range of 1 to 5 significant keywords per community. The major communities with significant keywords are depicted in Figure 4.4. Six of the 9 communities had one specific model of camera as a keyword (a620, s2, s1, s50, sd200, g2) and none had two camera model keywords. Again, across communities, there was no overlap between camera model keywords.

The keywords and communities from the $f = 0.8$ network are qualitatively better. In particular, the $f = 0.8$ network identifies a larger set of communities with at least one camera model keyword than the $f = 0$ network (6 vs. 2) without having more between-community overlap. One would expect that there are more than the 2 classes of substitutable camera models in the market represented by the $f = 0$ network. Furthermore, none of the communities in the $f = 0.8$ network had two different camera model keywords grouped together, while in the $f = 0$ network one of the communities had two fairly dissimilar camera models grouped together.

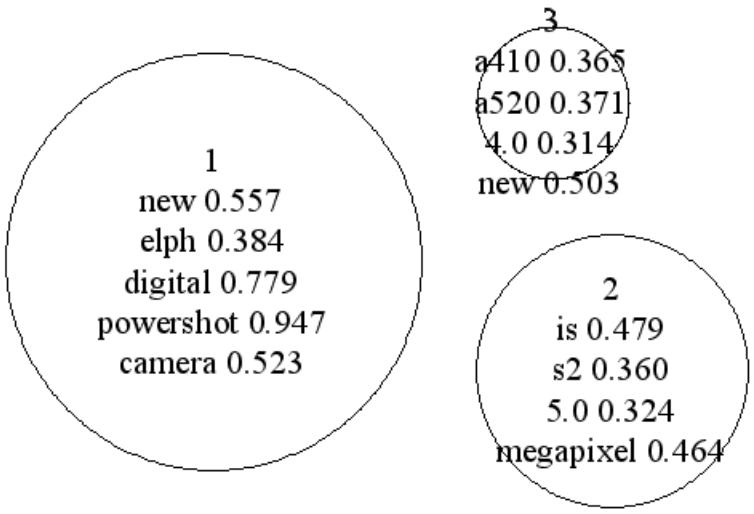


Figure 4.3: Canon Communities with No Edge Thresholding and with Unweighted Edges. The size of each circle is proportional to the number of nodes in each community. Significant keywords ($a \leq 0.01$, $p_c \geq 0.3$) and their associated p_c are listed in order of descending a (most overrepresented keyword first). Since the a values are all small, they are not shown. Placement of the communities in the figure is arbitrary.

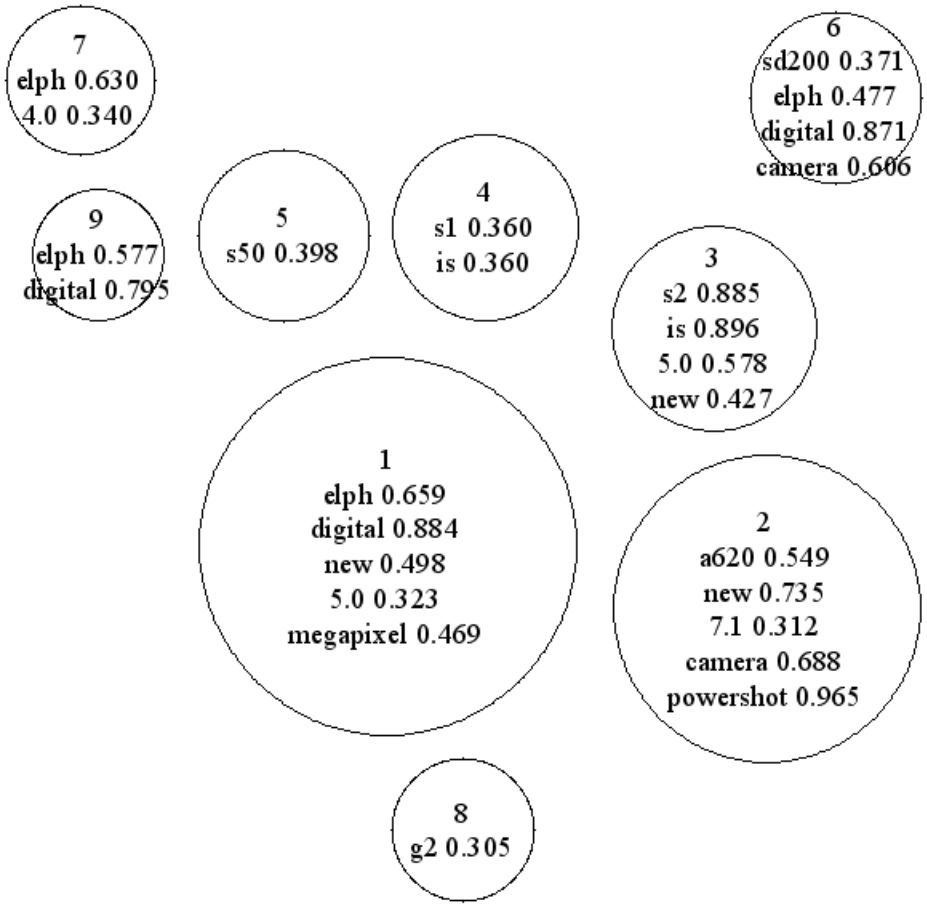


Figure 4.4: Canon Communities with 0.8 Edge Thresholding and with Unweighted Edges. The size of each circle is proportional to the number of nodes in each community. Significant keywords ($a \leq 0.01$, $p_c \geq 0.3$) and their associated p_c are listed in order of descending a (most overrepresented keyword first). Since the a values are all small, they are not shown. Placement of the communities in the figure is arbitrary.

4.3 Edge Weighting and Community Keywords

In addition to examining the effect of edge filtering on the quality of results, we also wanted to examine the effect of weighting edges by the number of shared bidders. We assessed the effectiveness of edge weighting qualitatively, in a similar manner as in the previous section. Again, we focus our analysis in this section on the Canon market. A qualitative analysis of the LCD market gave similar results and can be found in Appendix C.

We applied our solution to an un-edge-filtered, weighted Canon network and compared the results to those for the un-edge-filtered, unweighted network in the previous section (see Figure 4.3).

For the weighted network, there were 13 communities ranging in size for 3 to 757. Of these, 9 were major communities. We applied keyword extraction with the same parameters and examined the keywords. All 9 of the major communities had significant keywords, with a range of 1 to 5 significant keywords per community. These major communities are shown in Figure 4.5. Three of the 9 communities had exactly one camera model keyword (s2, a410, a520) and one had two (a620 and a610). Again, across communities, there was no overlap between camera model keywords.

The performance is similar to that of the unweighted Canon network. However, one could argue that it is slightly better for two reasons. First, the weighted network identifies a larger set of communities with at least one camera model keyword than the unweighted network does (4 vs. 2). In addition, the grouping of a620 and a610 in the same community in the weighted network is somewhat more justifiable than the grouping of a410 and a520 in the same community in the unweighted network, because the a610 and a620 models were released at the same time to replace an older model.[4] Nonetheless, the a620 and a610 also differ in their resolution, so they are not ideal substitutes.

[4]Information obtained from a review of the Canon a620 (http://www.dpreview.com/reviews/canona620/).

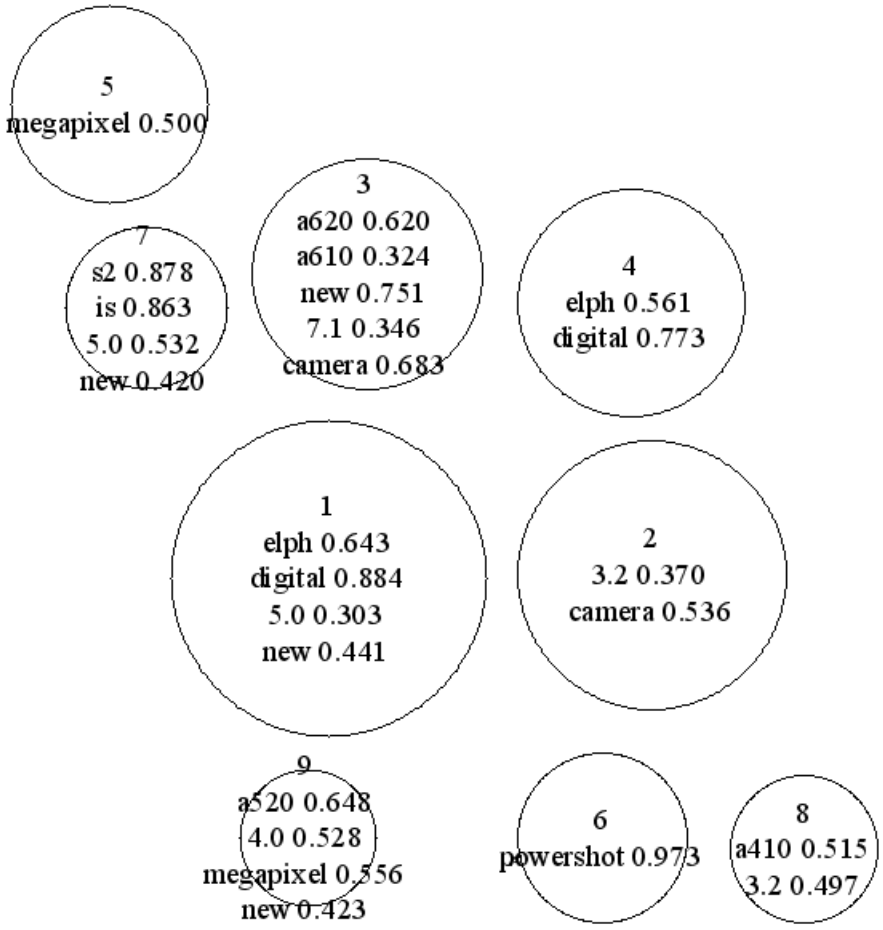


Figure 4.5: Canon Communities with No Edge Thresholding and with Weighted Edges. The size of each circle is proportional to the number of nodes in each community. Significant keywords ($a \leq 0.01$, $p_c \geq 0.3$) and their associated p_c are listed in order of descending a (most overrepresented keyword first). Since the a values are all small, they are not shown. Placement of the communities in the figure is arbitrary.

4.4 Qualitative Assessment of Our Solution with Edge Filtering and Edge Weighting in Conjunction

As a final step, we applied edge filtering and edge weighting in conjunction and assessed the results. For each of our markets—Canon and LCD—we generated a network that used both price threshold edge filtering (at $f = 0.8$) and edge weighting and applied our solution. We applied keyword extraction with the same parameters and examined the keywords. We discuss results for the Canon market first and then turn to the LCD market.

For the Canon market with both edge filtering and edge weighting, there were 25 communities ranging in size from 3 to 339. Of these, 15 were major communities. Twelve of the major communities had significant keywords, with a range of 1 to 6 significant keywords per community. These major communities are shown in Figure 4.6. Ten of these major communities had exactly one camera model keyword (sd500, a620, s2, a70, a520, s50, sd200, s80, s410, s110) and two communities had two camera keywords (sd400 and sd450; pro1 and g6). Again, there was no overlap of camera models between communities.

The pairs of camera keywords that were grouped in the same communities (sd400 and sd450; pro1 and g6) are natural substitutes. In particular, the sd400 is the predecessor to the sd450, and the two camera models are extremely similar—they share the same resolution and memory card, with the major difference being a slightly larger LCD screen on the sd450.[5] The substitutability of the pro1 and g6 is not quite as strong. However, they are 2 of the 6 "high-end" Powershot digital cameras that Canon offers and both take the same type of memory card (compact flash).[6]

This version of the network identified the most different camera model keywords (14). We wanted to determine the percentage of all Canon camera models in the market that our method identified. Ideally, we would want this percentage to be high. Unfortunately, there is no direct way to determine the total number of different camera models in the market because the lowest level of the current eBay hierarchy is by brand and not model.

To determine a plausible camera model candidate set, we obtained a list of current Canon digital cameras.[6] There were a total of 23 models; the name for 15 of these 23 models appeared at least once in the title of an auction in the Canon market data set.[7] The 8 models that did not appear in our data set tend to be newer camera models.

[5] One review (http://www.dpreview.com/reviews/canonsd450/) termed the sd450 a "fairly minor upgrade" to the sd400.

[6] Information obtained from the official Canon Powershot website (http://www.powershot.com).

[7] The 15 models that appeared at least once in our data set are (number of auctions matching in parentheses): a620 (578), sd550 (446), a610 (439), s2 (424), sd500 (417), a520 (345), sd450 (322), sd400 (294), a410 (235), s80 (206), sd30 (165), s70 (94), g6 (91), pro1 (85), sd430 (22).

Of the 15 current model keywords that are present in our data set, 9 of the model keywords (60%) were identified by our method. We examined the number of auctions in our data set matching each current model keyword and compared keywords that were identified to those that were not. We found that the 9 current-model keywords that were identified had a greater average number of auctions matching than the 6 current-model keywords that were not identified, although this difference was not significant (307 to 234, single-tailed T $a = 0.22$).

As noted earlier, our method identified a total of 14 model keywords—the remaining 5 keywords that our method identified correspond to models that were discontinued and thus no longer listed on the official Canon website.

Taking these results together, the application of our method to an edge filtered, edge weighted Canon network thus produced a community segmentation that was clearly superior to ones from edge filtering or edge weighting alone. In addition, the method was able to identify more than half of the current Canon models present in the eBay market and identified several discontinued Canon models in the market as well.

For the LCD market with both edge filtering and edge weighting, there were 42 communities ranging in size from 3 to 973. Of these, 12 were major communities. Ten of the major communities had significant keywords, with a range of 1 to 7 significant keywords per community. These major communities are shown in Figure 4.7. Eight of these 10 communities had a size (15", 17", 19", 20", 24") and/or a specific model number (e173fp, 2005fpw, 2405fpw) as a keyword. Importantly, no communities had more than one size or more than one model number. This property is desirable because it is unlikely that different sizes or models of monitors serve as good substitutes.

The communities are not quite as distinct as in the Canon market—for example, there are two communities with e173fp as a keyword and also two communities with 15" as a keyword. Thus, there is some evidence of oversegmentation. Nonetheless, when taken together, the keywords for the 12 major communities do seem to encapsulate the significant areas of the market—for example, all major monitor sizes are represented in at least one of the major communities.

An interesting anecdote is that examining the individual auctions in the LCD community corresponding to 24" 2405fpw Dell monitors (community 6 in Figure 4.7) revealed one case where our method correctly grouped an auction with similar other auctions even though the seller had listed it in an incorrect category. In this case, the seller listed the item in the 19-inch Dell monitor category (see Figure 4.8), but our method nonetheless grouped it with other 24" monitors of the same model.

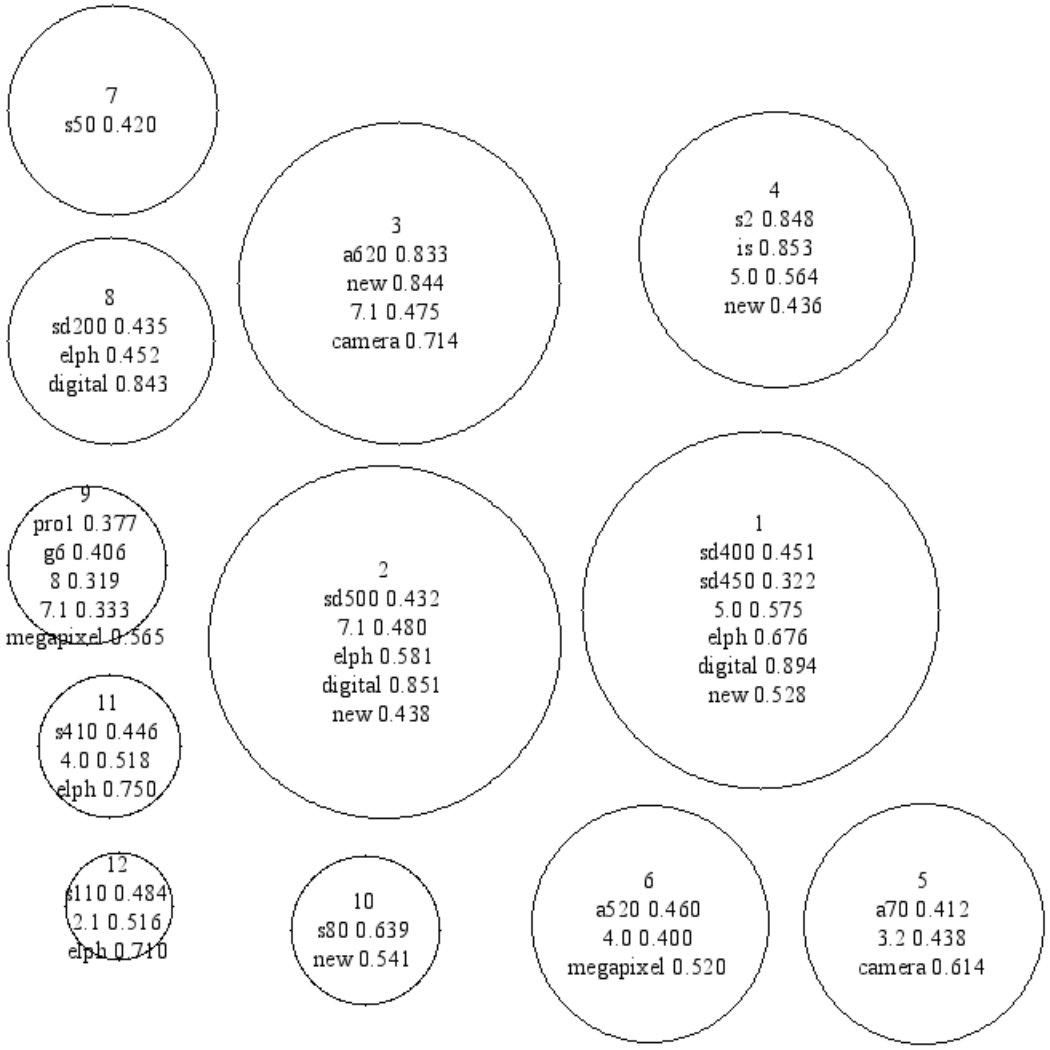


Figure 4.6: Canon Communities with 0.8 Edge Thresholding and with Weighted Edges. The size of each circle is proportional to the number of nodes in each community. Significant keywords ($a \leq 0.01$, $p_c \geq 0.3$) and their associated p_c are listed in order of descending a (most overrepresented keyword first). Since the a values are all small, they are not shown. Placement of the communities in the figure is arbitrary.

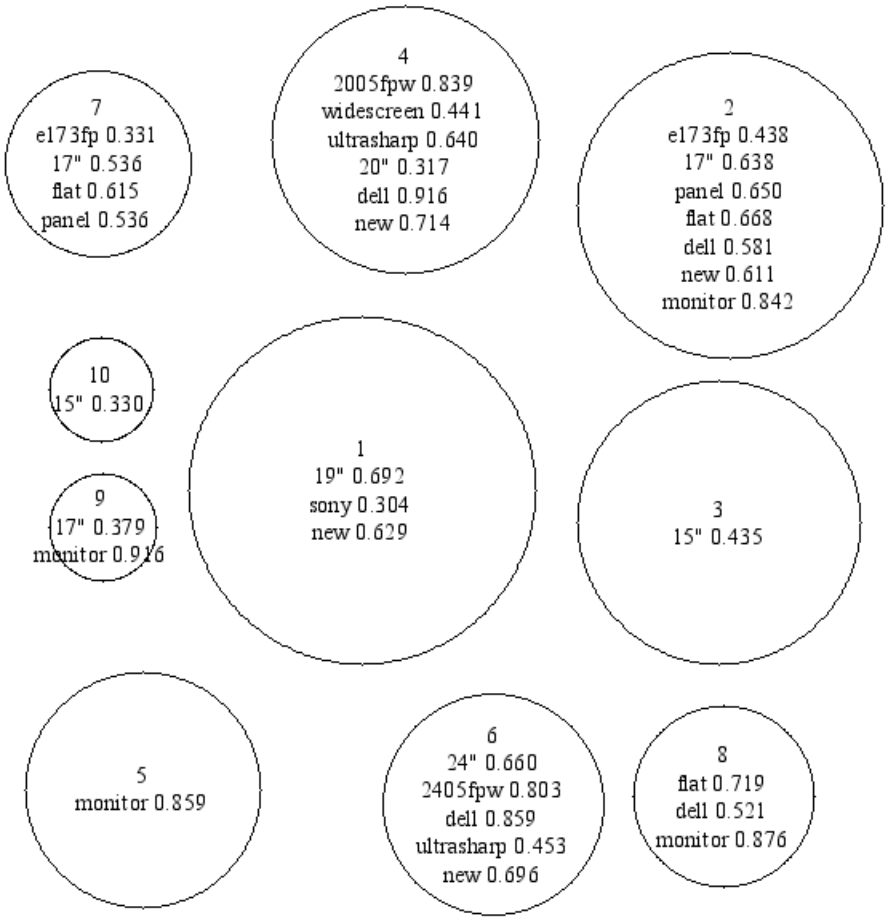


Figure 4.7: LCD Communities with 0.8 Edge Thresholding and with Weighted Edges. The size of each circle is proportional to the number of nodes in each community. Significant keywords ($a \leq 0.01$, $p_c \geq 0.3$) and their associated p_c are listed in order of descending a (most overrepresented keyword first). Since the a values are all small, they are not shown. Placement of the communities in the figure is arbitrary.

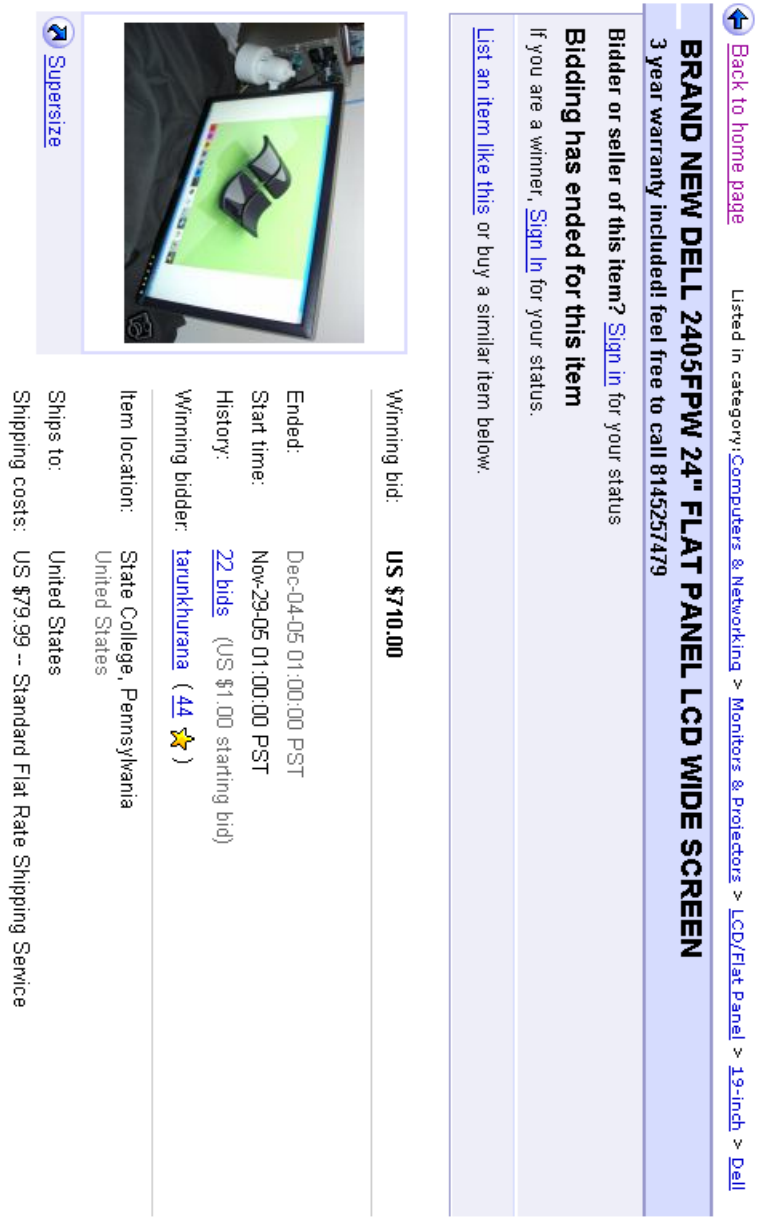


Figure 4.8: A Seller-Miscategorized Monitor Listing. Our method grouped the item (id=8730606839) with other items of the same model (24" 2405fpw) even though the seller listed it in a category for 19" monitors.

4.5 Discussion

In both the Canon and LCD markets, the results of applying our method to an edge-weighted and price-thresholded network produced qualitatively good community segmentations and keywords. In the Canon market, for example, the method segmented the market into well-defined, non-overlapping communities characterized by keywords for camera model and resolution. The results are particularly striking because our method was given absolutely no *a priori* information about the type of goods in the market—instead, using bidder behavior alone, our method was able to automatically identify the most relevant 20-30 item-defining keywords in the Canon market from a global set of 899 and categorize auctions based on these keywords.

Despite the relatively good performance on the two data sets, there are a number of limitations to the method. One limitation is the dependence on assumptions about bidder behavior, and another is the data loss that occurs during network construction because we only look at the largest MCC. We will now discuss each of these in turn.

Our method is extremely dependent on assumptions about bidder behavior. As noted before, we assume that bidders tend to bid in auctions that are for similar goods. In general, this assumption is not problematic. However, in certain markets where there are strong complementarities of non-substitutable goods, this assumption might not hold true. In particular, if, in these markets, bidders tend to bid more in auctions that have non-substitutable complementary goods than in auctions that have substitutable goods, then our method might not perform as well.

Generating a network with edges based on shared bids across auctions from the same bidder also introduces an implicit temporal bias to the network. If the time period of interest for a type of item for each bidder is bounded by some value t,[8] then in order to draw an edge between two auctions, the auctions must be no further than t apart in time. Temporal fluctuations in bidder volume or behavior patterns might bias the network in an undesirable manner. In addition, auctions with a longer duration might be better connected in the network simply because there is a longer time period during which bidders can generate edges to them. The effect (if any) of the time range of the data set used to generate the network on the results also remains to be determined.

Another limitation of the method is that the largest MCC to which we apply community detection represents only a subset of all of the auctions in the data set. This is because we

[8] We define the time period of interest as the length of time during which a bidder is actively trying to buy an item. It is hard to imagine that bidders will participate indefinitely on eBay to buy an item. Eventually, if they are unsuccessful in winning the item, they will turn to other markets.

can only draw an edge between two auctions if they have a shared bidder—not all auctions will have shared bidders with other auctions. The Canon set consisted of 6717 auctions, 4308 of which had at least one bidder; the largest MCC without edge filtering had 3173 nodes. The LCD set consisted of 11782 auctions, 8288 of which had at least one bidder; the largest MCC had 6024 nodes.

In both of our markets, even after edge filtering at $f = 0.8$, the largest MCC still had over half of the auctions in the data set. Thus, we still are applying community detection to a large fraction of the market. However, in filtering out auctions, we might lose some of the smaller communities of goods.

We discuss future work and applications in the concluding chapter of the thesis.

Chapter 5

The Complements Problem

Having discussed our solution to the substitutes problem, we now extend our work to a related problem—detecting complementary goods. We begin this section by defining the problem. Then, we discuss the additional data sets we collect in order to evaluate our methods. Next, we discuss our preliminary solution to the problem and present results. We conclude with a discussion of the results and the current limitations to our method.

5.1 Problem Definition and Motivation

Two goods are complementary if the value to a bidder of having the pair is greater than the sum of the values of having each separately. In contrast to substitutes, if the price of one of a pair of complementary goods goes up, the demand for the other complementary good may go down. As noted earlier, an example of a pair of complementary goods is a digital camera and a memory card. Having a memory card for the digital camera increases the value a user gets from the camera. If the price of memory cards were to go up, the demand for cameras would likely go down; similarly, if the price of cameras were to go up, the demand for memory cards would likely go down.

In this chapter, we investigate whether we can identify complementary goods on eBay. More specifically, we propose and assess methods for identifying pairs of goods that bidders often buy together. We assume that buying goods together is evidence for complementarity.[1] As noted earlier, a method of detecting complementary goods could be applied to refining "recommender systems" that give bidders suggestions on goods to purchase based on their

[1] As in the substitutes section, we also assume that bidders generally will use the same user name for their transactions.

past purchases. Complementary goods can occur in sets larger than two, but as a starting point, we only consider complementary pairs of goods.

One should note that an approach to the complements problem requires a solution to the substitutes problem, since one needs to know the item types of interest before being able to assess complementary relationships. For example, if we are given a data set containing cameras and memory cards and want to assess complement relationships between different types of cameras and different types of memory cards, we would need to first determine the major substitutes communities—that is, what the different types of cameras and memory cards are. We propose a method for detecting complements if the substitute communities of interest are known and also propose a method for detecting complements if the substitute communities of interest are not known.

5.2 Additional Data Sets

In order to better assess our methods, we collected two additional data sets. The first set (Secure Digital) consists of all auctions matching "Secure Digital" in the Secure Digital memory card category over a period from January 10, 2006 to January 25, 2006. The second set (Compact Flash) consists of all auctions matching "Compact Flash" in the Compact Flash memory card category over the same time period. The time period was the same as the time period for our Canon data set.

As discussed above, we have reason to believe that memory cards and cameras have complementary relationships. Furthermore, different models of camera require specific formats of memory cards—either compact flash or secure digital. Thus, we can assess the effectiveness of our methods by comparing the strength of complementary relationships detected for each camera model and the two types of memory cards.

The Secure Digital set consisted of 12803 auctions. 7348 of the auctions had at least one bidder (57%). 7289 of the auctions sold (57%); the remainder may have failed to meet a reserve price or may have been cancelled by the seller. 9092 (71%) of the auctions did not have a buy-it-now option and 107 (0.83%) had a reserve price. As with the previous sets, most bidders win only one auction. 5807 of the 6376 unique winning bidders won only one auction (91%).

The Compact Flash set consisted of 4910 auctions. 2718 of the auctions (55%) had at least one bidder. 2646 of the auctions (54%) sold. 2404 (49%) of the auctions did not have a buy-it-now option and 125 (2.5%) had a reserve price. 2076 of the 2646 unique winning bidders won only one auction (90%).

We also examined the number of bidders who won items in two different markets and found that these numbers were low: 75 bidders won an item in both the Canon and Secure Digital market, 45 bidders won an item in the both the Canon and Compact Flash market, and 92 bidders won an item in both the Compact Flash and Secure Digital markets.

5.3 Our Solution

As noted above, to address the complements problem one must know the communities of items that one wants to assess complementary relationships between. For example, in our data set, natural communities of items to examine would be different models of cameras and different types of memory cards. In this section, we first present a method for detecting complements if the substitute communities of interest are known. Then, we present a method for detecting complements if the substitute communities of interest are not known.

5.3.1 Known Communities of Interest

The first case we consider is if the communities of interest are known. One situation where this case applies would be if we had already identified camera models and wanted to see if there were complementary relationships with the different memory card types. In this case, we simply need to define a means to evaluate the strength of the complementary relationship between two communities of goods c_1 and c_2. Intuitively, if the two communities have a large number of shared winning bidders (bidders who win items in both c_1 and c_2), then it is likely that the two communities have high complementarity. We propose two different but related ways to measure complementarity: $comp(c1, c2)$ and $comp_T(c1, c2)$. We define $comp(c1, c2)$ as:

$$comp(c1, c2) = max(cpct(c1, c2), cpct(c2, c1)) \tag{5.1}$$

where $cpct(a, b)$ is the number of distinct winning bidders in a that also win at least one auction in b divided by the total number of distinct winning bidders in a. In this definition, we use percentages (rather than just taking absolute numbers) because it makes sense to normalize by community size. The fact that there are 10 shared winning bidders between two communities is extremely significant if the communities are each size 10 but much less significant if the communities are each size 10000.

By the definition above, $comp(c1, c2)$ is symmetric but $cpct(c1, c2)$ is not. The definition of $cpct(c1, c2)$ captures the idea of asymmetric complementarities. In the digital camera and memory card domain, we might expect in general that $cpct(c, m) > cpct(m, c)$ where

c is a camera community and m is a memory card community, since most purchasers of a new camera would want to also buy a new memory card, but not every purchaser of a new memory card would also want to buy a new camera (memory cards can be used in other electronics, such as PDAs). $comp(c1, c2)$ takes the maximum of these asymmetric complementarities. Thus, $comp(c1, c2)$ represents the strongest of the two (potentially asymmetric) complementarity relationships between any two communities.

In addition, $comp(c1, c2)$ only measures complementarity between different communities. Some communities might be complements with themselves—for example, some bidders might gain added marginal utility from buying multiple copies of a memory card. Exploring asymmetric complementarities and within-community complementarities would be an interesting area for future work, but $comp(c1, c2)$ is a reasonable starting point.

In some cases, particularly when community sizes are small or when there are only weak complementary relationships in the market, $comp(c1, c2)$ might not be informative. In particular, when there are very few auctions with shared winning bidders, $comp(c1, c2)$ will be 0 over much of its domain. As we saw in the previous section, there is in fact a sparsity of shared winning bidders in our data set. We thus also propose a less restrictive, tunable version of $comp(c1, c2)$, which we call $comp_T(c1, c2)$. We define $comp_T(c1, c2)$ as:

$$comp_T(c1, c2) = max(cpct_T(c1, c2), cpct_T(c2, c1)) \tag{5.2}$$

where $cpct_T(a, b)$ is the number of distinct winning bidders in a that also place a bid that is at least a fraction T of the closing price of an auction in b divided by the total number of distinct winning bidders in a. The intuition here is similar to that found in the substitutes chapters for price threshold edge filtering. A high bid fraction signifies interest in an item. Note that $comp_1(c1, c2)$ is essentially equivalent to $comp(c1, c2)$—the only difference is that $comp_1(c1, c2)$ includes bids that tied the closing price but lost.

Having defined these metrics, assessing complementary pair relationships for a set of communities simply involves evaluating the metric for each pair.

5.3.2 Unknown Communities of Interest

The above discussion assumes that the item communities of interest are known. This assumption might not always be true. Imagine, for example, that we are given a combined market of Canon, Secure Digital, and Compact Flash data (without any hierarchical or community information, such as camera models) and want to determine the major complementary relationships between communities of identical goods that are not known beforehand. In this subsection, we present a method that, given a large, heterogeneous market,

can automatically group and suggest communities that might have complementary relationships. This method requires less *a priori* knowledge about the data set than the method described in the previous subsection.

Our approach involves three steps. First, we group individual goods that likely have complementary relationships. Next, we examine the major communities of complement candidates ("complement supercommunities") from the first step and try to subdivide each of them into distinct categories. Finally, having identified the subcommunities ("substitute subcommunities") of interest within each larger complement supercommunity, we then apply the methods in the previous subsection to these subcommunities. We discuss the details of each step in turn.

1. We group individual goods that likely have complement relationships into complement supercommunities. In this step, we would want to group items by some measure of shared winning bidders. If the same bidder wins two different items, then we have reason to believe that a complement relationship between the two items exists. The most obvious way of grouping items by shared winning bidders is to generate an auction network $W_0 = (V, E)$ where an edge e is drawn between two auctions v_1 and v_2 iff the same bidder won the two auctions. Unfortunately, with this approach, there is very little connectivity in the network. For example, if we refer to Table 4.1, we see that the largest MCC for the Canon network at edge-filter $f = 1$ is only 22, and the shared winning bidder network is even more restrictive in edge generation.[2] Thus, instead of having a largest MCC with a majority of nodes, we instead end up with many disconnected smaller MCCs.

   To solve this connectivity problem, we propose overlaying our winner information on a more connected network when applying community detection. We generate a network S_0 using edge weighting and edge filtering as in the substitutes problem but apply a modified version of the Greedy Q community detection algorithm, which we term the complements supercommunity detection algorithm. Rather than initializing the Greedy Q algorithm with each node in its own community, we instead initialize communities to be the individual MCCs in the auction network W_0, as defined above. Running community detection in this way should emphasize communities containing

[2] As noted earlier, the network at $f = 1$ generates edges between two auctions if the same bidder bid in both and placed a bid that was 100% of the closing price in each. This network is quite similar to the shared winning bidder network; however, in the $f = 1$ network bids that tied the maximum but lost (eBay breaks ties in favor of the earlier bidder) are still considered in edge generation, while in the winning bidder network these bids are not considered.

complementary goods (since auctions won by the same bidder fall in the same community) and also substitutes for those complementary goods (since we begin with the original substitutes network).[3]

2. We identify the substitute subcommunities within each larger complement supercommunity. From the previous step, we should now have supercommunities with a mixture of complementary goods and their substitutes. In this step, we want to separate each complement supercommunity into substitute subcommunities. Thus, for each supercommunity, we run the standard Greedy Q community detection algorithm on the corresponding subgraph from S_0.[4]

3. We determine the complementary relationships between the substitute subcommunities of goods identified. Once we have identified the substitute subcommunities of interest, we can simply apply the methods from the previous subsection to them. In particular, we can generate a $comp(c1, c2)$ matrix for all of the subcommunities in each supercommunity.

Note that in markets where no strong complements exist, the method above reduces to a hierarchical application of the substitutes community detection algorithm. The difference between our method and a hierarchical application of the substitutes community detection is that complementary goods (if they exist) should tend to fall in the same supercommunity after the first step.

5.4 Results

In this section, we present preliminary results for each of the methods outlined in the previous section.

5.4.1 Known Communities of Interest

We were interested in determining if our $comp(c1, c2)$ and $comp_T(c1, c2)$ metrics can distinguish between compact flash and secure digital complements for the different model communities of Canon camera identified in the previous chapter (see Figure 4.6). We chose

[3] An alternate way to get a similar result might be to give edges made by shared winners in S_0 an additional weight w and then run the standard Greedy Q algorithm. Adding the weight also forces auctions won by the same bidder to tend to cluster in the same community.

[4] We use S_0 because in this step we are interested in finding the communities of identical goods and not complements.

to examine these relationships because we have reason to suspect what the correct associations should be, making it easier to interpret results. As noted above, Canon digital cameras take either compact flash or secure digital memory cards. Of the 12 camera model communities, 7 were secure digital cameras and 5 were compact flash cameras.

We examined $comp(c1, c2)$ and $comp_T(c1, c2)$ for 14 communities. Twelve of these communities were the camera communities from the previous chapter. In addition, we generated one compact flash community and one secure digital community. Each of the memory card communities contained a random 10% sample of successfully sold auctions in their respective markets.[5] We then generated 14x14 matrices for $comp(c1, c2)$ and $comp_T(c1, c2)$ (for $T = 0.9, 0.7, 0.5$, and 0). An entry i, j in the matrix represents the value of the $comp(i, j)$ or $comp_T(i, j)$. Since the functions are symmetric, the matrices are necessarily symmetric as well.

In Figure 5.1, we display the complement function values for the compact flash and secure digital memory card communities in relation to the camera communities. Most of the values for $comp(c1, c2)$ are 0—that is, there were no shared winners between most of the community pairs. For 4 of the 7 secure digital camera communities, there was a nonzero $comp(c1, c2)$ value between the camera community and the secure digital community. For 1 of the 5 compact flash camera communities, there was a nonzero $comp(c1, c2)$ value between the camera community and the compact flash community. Importantly, there were no false positives—none of the secure digital camera communities had nonzero $comp(c1, c2)$ with the compact flash community, and none of the compact flash camera communities had nonzero $comp(c1, c2)$ with the secure digital community. While $comp(c1, c2)$ does not result in any false positives, it only associates 5 of the 12 camera communities with the expected memory card model. The remaining 7 communities had complement function values of 0 with both memory card types. It is possible that no complement relationships exist for the other 7 communities; however, it is also possible that $comp(c1, c2)$ is too restrictive and thus fails to identify complement relationships that do exist.

We next assessed whether we could identify more associations by relaxing the definition of $comp(c1, c2)$. At the least restrictive value of T, $comp_0(c1, c2)$ the function is able to associate 8 of the 12 camera communities with their correct memory card type, which is almost twice as many associations identified by $comp(c1, c2)$. In this process, however, one compact flash camera community is potentially misclassified, since it has similar $comp_0(c1, CF)$ and

[5]We did not generate substitute communities for the memory card markets because we simply wanted to test for complements between cameras and memory card types in general. For future work, it would be interesting to examine whether we can detect stronger complements between some models of camera and some subtypes of memory card—for example, if higher resolution cameras have strong complementary relationships with higher capacity memory cards.

$comp_0(c1, SD)$ values.

The data suggest that there is some between-camera-community complementarity—that is, there are bidders who win multiple cameras from different communities. Figure 5.2 depicts the full 14x14 density plot for $comp(c1, c2)$. As we see, only 2 of the 4 camera communities with memory card $comp(c1, c2)$ relationships had their strongest $comp(c1, c2)$ value with the memory card community. The other two camera communities had stronger complement relationships with another camera community.

As we relax our definition of $comp(c1, c2)$, the between-camera-community values of $comp_T(c1, c2)$ begin to dwarf the camera-memory card values of $comp_T(c1, c2)$. This fact becomes evident in Figure 5.3, which depicts a 14x14 density plot for $comp_0(c1, c2)$. While 8 of the 12 camera communities are associated with the correct memory card type, none of these relationships had the highest value of $comp_0(c1, c2)$ for their row.

Interestingly, there seems to exist stronger complementarity between secure digital cameras and secure digital cards than between compact flash cameras and compact flash cards. Further investigation is needed to determine if this phenomenon is due to an intrinsic property of the goods and market or if it is due to some bias in our complementarity metric.

Together, these results suggest that while relaxing $comp(c1, c2)$ can result in the identification of more associations, there are potential trade-offs in increased misclassification and noise.

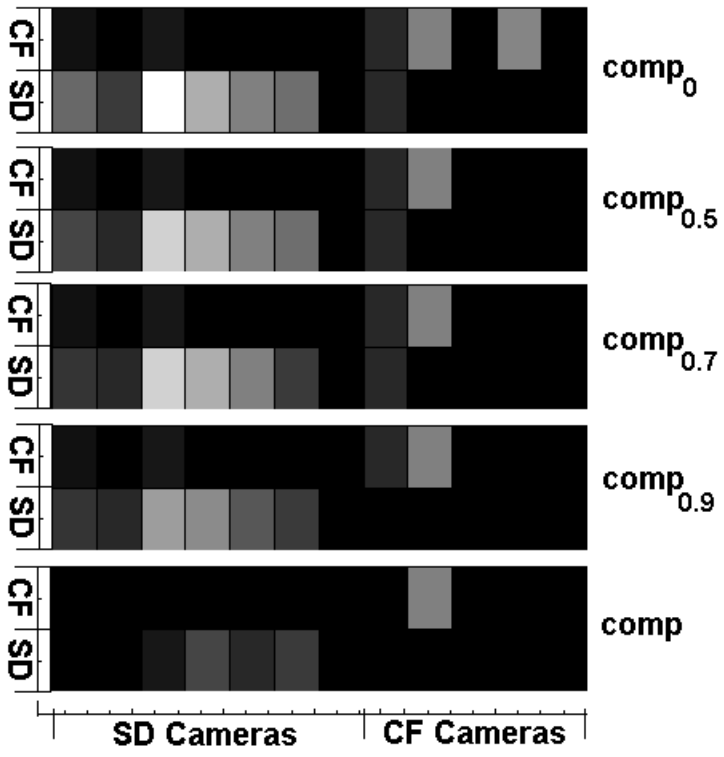


Figure 5.1: 2x12 Density Plots of Different Complementarity Function Definitions. Each density plot shows the complementarity function values between the two memory card types (CF and SD) and the 12 camera communities, 7 of which use SD memory cards and 5 of which use CF memory cards. Relaxing the definition of complementarity function reveals more correct associations. Brighter boxes indicate a higher value.

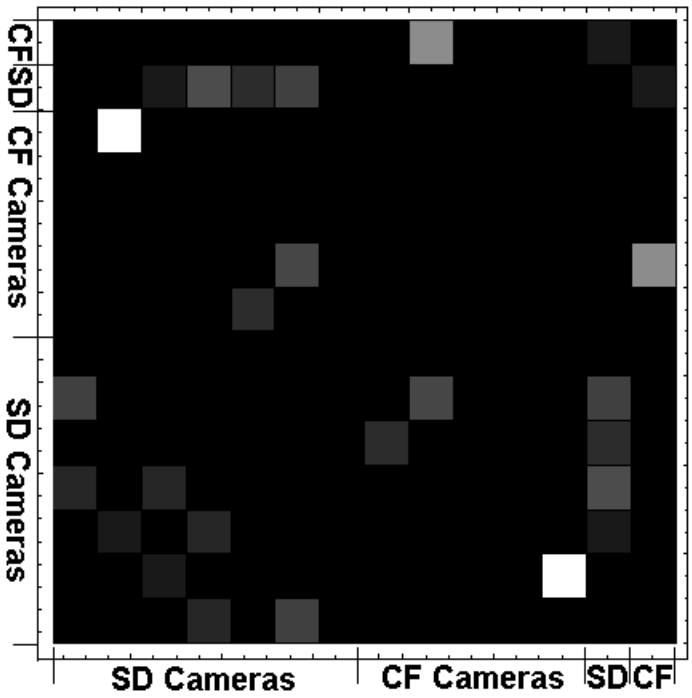


Figure 5.2: 14x14 Density Plot of $comp(c1, c2)$. Brighter boxes indicate a higher value.

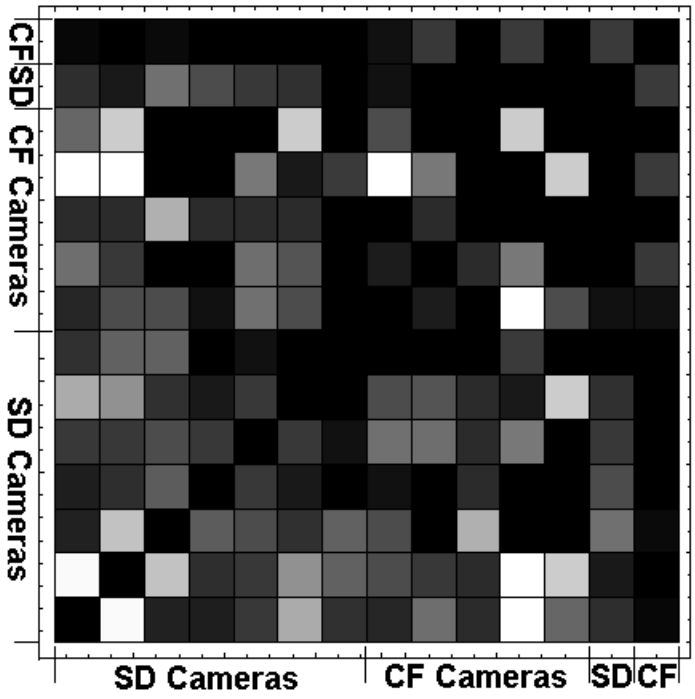


Figure 5.3: 14x14 Density Plot of $comp_0(c1, c2)$. Brighter boxes indicate a higher value.

5.4.2 Unknown Communities of Interest

In the previous subsection, we discussed the results pertaining to our method of assessing the complementarity relationship between known communities of identical goods. We now turn to evaluating our method for identifying major complementary relationships between communities of identical goods that are not known beforehand.

As noted before, our method involves three steps. The first step aims to group goods with strong complementary relationships into the same complements supercommunity. The second step aims to identify the categories of goods (substitutes subcommunities) that lie within each complements supercommunity. The third step then assesses the complementary relationships between these substitutes subcommunities.

To evaluate our method, we took a combined market of Canon, Secure Digital, and Compact Flash data. We wanted to see if the method would be able to separate Canon cameras by their complement relationships with memory cards. In other words, we wanted the see if the communities found after the first step would contain only one type of camera and associated memory card (either secure digital cameras and secure digital memory or compact flash cameras and compact flash memory, but not secure digital cameras and compact flash memory or compact flash cameras and secure digital memory.)

For the first step of our method, we generated the S_0 network for the combined data set using $f = 0.8$ and edge weighting by shared bidders. We then applied the modified complements supercommunity detection algorithm that initializes communities to W_0. For comparison, we also applied the standard community detection algorithm.

The complements community detection algorithm produced 44 communities ranging in size from 2 to 1881. The standard community detection algorithm produced 59 communities ranging in size from 2 to 3176. In order to qualitatively assess the groupings of camera and memory, we looked for the largest community in each set that had the keyword Canon at $p_c > 0.05$. The third largest community (size 1194) from the complements community detection algorithm and the second largest community (size 1705) from the standard community detection algorithm satisfied this criterion.[6] In the following paragraphs, we refer to the community from the complements algorithm as the complements supercommunity and the community from the standard algorithm as the standard supercommunity.

We then ran the second step of our method, applying standard community detection to each of these supercommunities with $a = 0.01$ and $p_c = 0.3$. We consider a community to

[6]The larger communities in the sets that did not satisfy the criterion seemed to be dominated by memory cards.

be major if it contains at least 5% of the auctions in its supercommunity.[7] The results are shown in Figures 5.4 and 5.5.

In the complements supercommunity, there were 7 major substitutes subcommunities with at least one significant keyword. Four of the subcommunities contained cameras that take secure digital memory (sd500, sd400, sd450, and sd550) and two of the subcommunities contained secure digital memory cards (512mb and 1gb/2gb). The last community (the smallest) contained Canon cameras of unspecified model. This result makes sense, since the community associates secure digital memory cards with secure digital cameras. Furthermore, the segmentation into substitutes is robust—first, there is no overlap between camera models and second, the division of memory cards by size (512mb and 1gb/2gb) is reasonable.

In comparison, in the standard supercommunity, there were 8 major communities with at least one significant keyword. Six of these communities contained cameras that take secure digital memory and one contained compact flash memory cards. The last community (the second smallest) contained cameras of unspecified model. From a complements perspective, this result is not as good as the complements supercommunity result for two reasons—it fails to associate secure digital memory cards with secure digital cameras and furthermore incorrectly associates compact flash memory cards with secure digital cameras. We note, however, that the segmentation into substitute communities is again quite qualitatively good.

We then applied the third step of our method to both sets of subcommunities. Since we have identified the communities, we generated matrices for $comp(c1, c2)$ for each set following the methodology in the previous subsection. The results are shown in Figures 5.6 and 5.7.[8] Not surprisingly, there was stronger complementarity in the complements supercommunity than in the standard supercommunity.

In summary, the above results provide preliminary evidence for the potential effectiveness of our method for automatically finding complementary groups of goods works. However, more analysis is needed. For example, we would need to examine in detail each of the other major complements supercommunities generated and compare them to the standard supercommunities.

[7] In the substitutes section we used 1%, but we raised the percentage slightly here because the supercommunities are smaller than those in the substitutes section.

[8] The two communities that did not have a specific camera model are not shown.

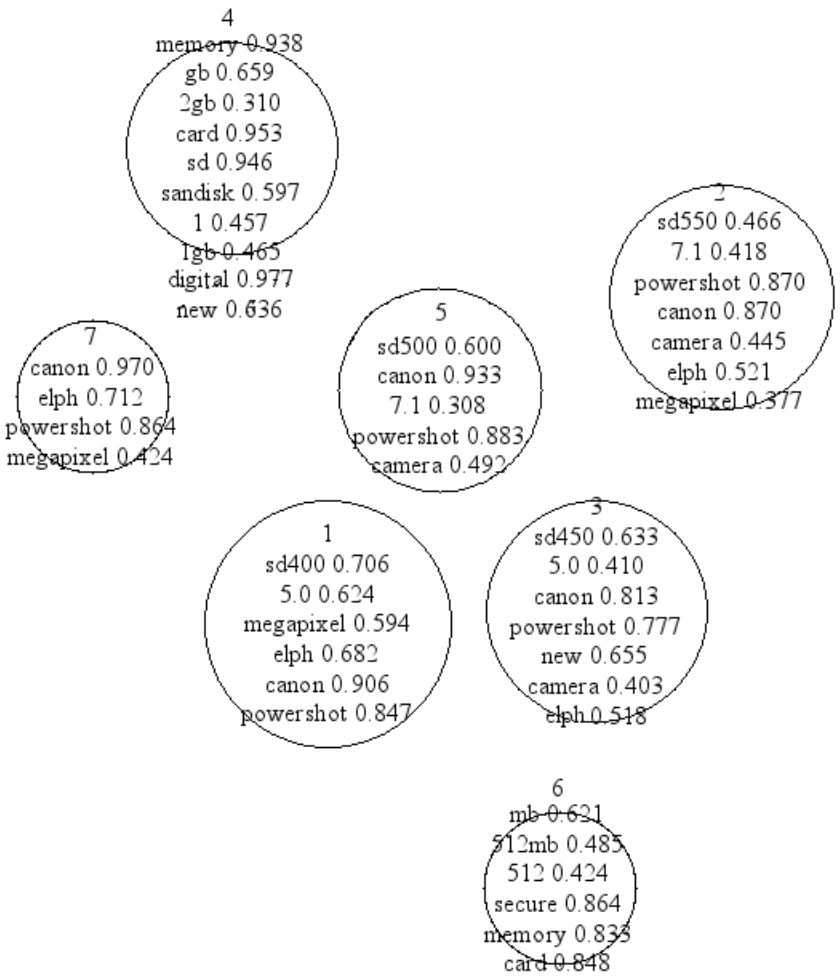


Figure 5.4: Subcommunities of the Complements Supercommunity with 0.8 Edge Thresholding and Weighted Edges. The size of each circle is proportional to the number of nodes in each community. Significant keywords ($a \leq 0.01$, $p_c \geq 0.3$) and their associated p_c are listed in order of descending a (most overrepresented keyword first). Since the a values are all small, they are not shown. Placement of the communities in the figure is arbitrary.

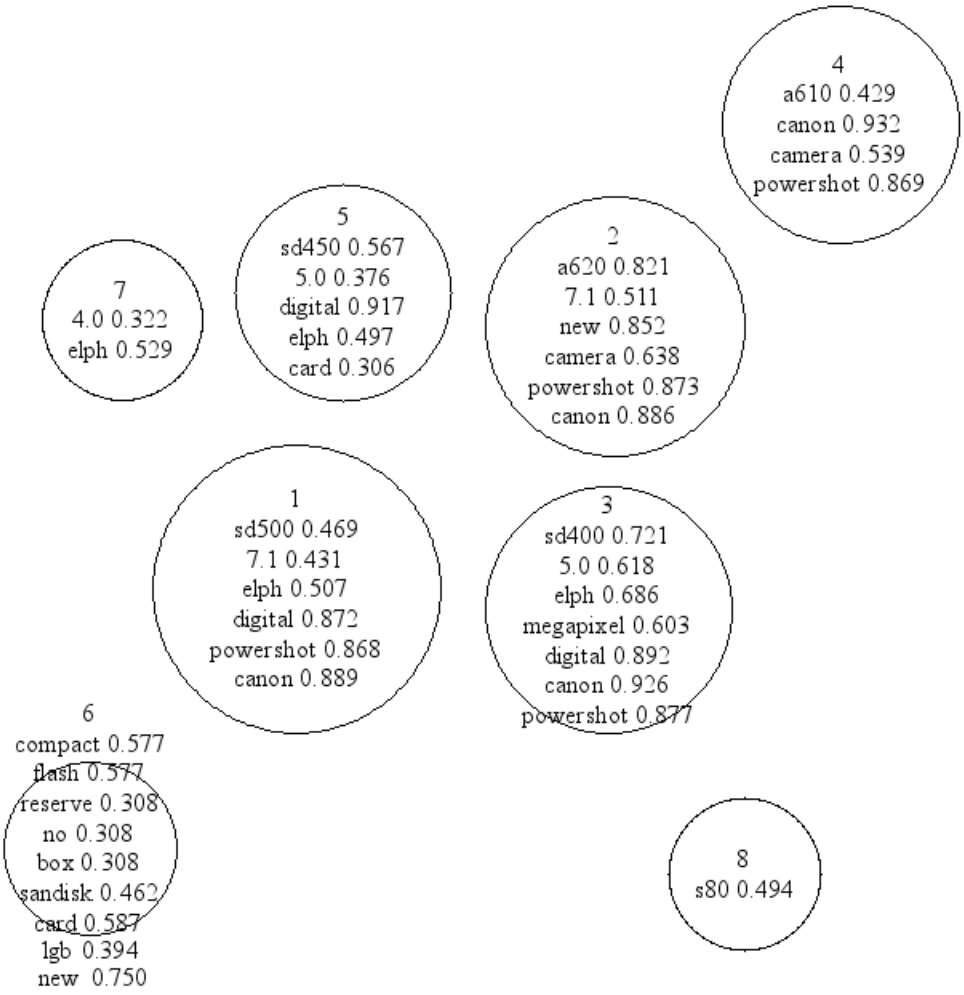


Figure 5.5: Subcommunities of the Standard Supercommunity with 0.8 Edge Thresholding and Weighted Edges. The size of each circle is proportional to the number of nodes in each community. Significant keywords ($a \leq 0.01$, $p_c \geq 0.3$) and their associated p_c are listed in order of descending a (most overrepresented keyword first). Since the a values are all small, they are not shown. Placement of the communities in the figure is arbitrary.

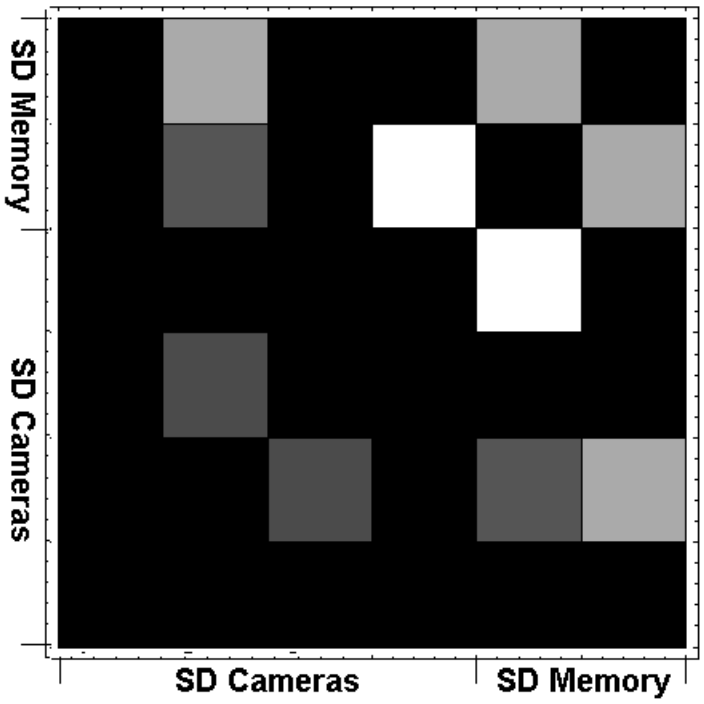


Figure 5.6: Density Plot of $comp_0(c1, c2)$ for Substitutes Subcommunities in the Complements Supercommunity. Brighter boxes indicate a higher value.

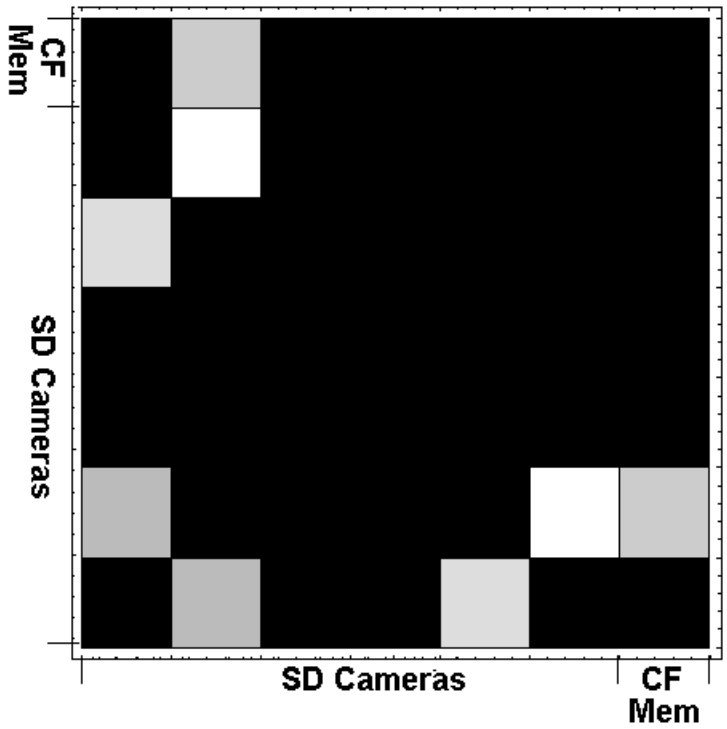


Figure 5.7: Density Plot of $comp_0(c1, c2)$ for Substitutes Subcommunities in the Standard Supercommunity. Brighter boxes indicate a higher value.

5.5 Discussion

In this chapter, we extended our work in the substitutes domain to the related problem of complements. We proposed a class of metrics to evaluate the complementarity between any two communities of goods. The metrics were able to correctly associate camera communities with their complementary memory card communities despite the relative sparsity of shared winning bidder information. In addition, we presented a method for automatically finding communities that exhibit strong complementarity relationships and provided preliminary evidence that it was effective.

Despite the generally good results, the methods we propose do have some limitations. As previously mentioned, one limitation of applying the metrics we propose is that they depend very strongly on having good substitutes communities. If substitutes communities are noisy (for example, if camera models with different memory types are grouped into the same community), then the metrics will not give good results.[9]

In addition, the metrics may have difficulty distinguishing between substitutes and complements, especially when we lower T in $comp_T(c1, c2)$. As we see in Figure 5.3, camera models had higher values of $comp_T(c1, c2)$ with other camera models (which are better substitutes) than with memory cards (which are better complements). At the most restrictive definition of $comp(c1, c2)$ this problem was less severe; however, there were also fewer camera-memory card associations identified.

As in the case of substitutes, our complements methods also have an implicit temporal bias. We only collected two weeks worth of data—if bidders do not try to win complementary items simultaneously (or in a short time range), then we will miss complement relationships.

We discuss future work and applications in the next and final chapter of the thesis.

[9]The fact that we were able to get reasonable complementarity results in this section is additional evidence that the communities identified by our substitutes algorithm are good.

Chapter 6

Conclusions

As online auction sites grow in size, it becomes important to have a method of organizing the goods available into reasonable categories. Auction sites are successful because they bring sellers and buyers together, and having a good item categorization structure makes it easier for buyers to find the items that they want. As we noted earlier, studies have demonstrated that the categorization needs to extend to the level of substitutable goods in order to be most effective.

In this thesis, we have proposed a method to solve the substitutes problem and demonstrated it to be effective. The method relies on a minimum of expert knowledge, which is important because the online auction markets list goods across a variety of domains. Additionally, we extend our work to the related problem of complementary goods.

There are three major sections in this chapter. The first section summarizes our major contributions. The second section lists areas where our methods can be improved. The final section contains a discussion of future research and applications.

6.1 Summary of Contributions

As we noted earlier, relatively little work has been done previously on the problems addressed in this paper. Indeed, community detection algorithms have been applied only once in the literature to networks derived from online auction markets. In this section, we highlight the major contributions of this thesis to the substitutes and complements problems.

6.1.1 The Substitutes Problem

The key insight behind our solution to the substitutes problem is one about bidder behavior—namely, that bidders tend to bid in auctions with substitutable items. We generate a novel auction network that makes use of this insight, which we believe gives it an advantage over other methods, such as text-based clustering.

Beyond the overall solution, the major contribution of our work lies in the way we refined our auction network. We found that we could improve our results dramatically by filtering edges, and to a lesser extent by weighting edges. We proposed two ways of filtering edges to remove edges that were uninformative with regards to substitute community structure. We demonstrated that both of these filtering methods increased the modularity score of the network; furthermore, edge filtering by price thresholding resulted in qualitatively better substitute communities. We also showed that edge weighting by the number of shared bidders resulted in qualitatively better substitute communities.

When we combined price-threshold edge filtering with edge weighting and applied our solution, we found that we were able to generate reasonable communities of substitutable goods in both the Canon and LCD markets of eBay.

6.1.2 The Complements Problem

One major contribution to solving the complements problem was our formulation of metrics for determining the strength of a complement relationship between any two known communities. We found that data on complement behavior was relatively sparse in the markets we examined, despite the fact that we had reason to suspect the existence of complement relationships. A low percentage of bidders won multiple items in the Canon, Secure Digital, and Compact Flash markets; as a result, the first formulation of our metric did not identify many of the expected relationships. We proposed ways of relaxing our metric so that we could detect complements even when there were not shared winners and demonstrated that these relaxed metrics perform better, although there is a trade-off with increased noise.

The second contribution to solving the complements problem was our method for automatically detecting and grouping substitute communities that have strong complement relationships even when no substitute or hierarchical structure information is known beforehand. We presented preliminary results suggesting that our method could be effective.

6.2 Improvements to Our Solutions

Despite the generally positive results outlined above, there are still several areas where our methods could be improved. In this section, we discuss possible improvements to our methods for the substitutes and complements problems.

6.2.1 The Substitutes Problem

In the substitutes domain, a better community detection algorithm would obviously improve results. While the greedy modularity algorithm is widely used, there has not been work done on proving a performance bound. Assessing alternate algorithms for maximizing Q, such as simulated annealing, would be worthwhile. In addition, the formal properties of Q are largely unexplored. While Q works well in practice, it is possible that an alternate metric for community detection might provide better results.

There is also room to improve the keyword extraction algorithm. One could improve keywords extracted by looking at auction listing text beyond the title. One could also implement a more sophisticated algorithm, such as one that looks at n-grams rather than just single words.

Finally, work can be done to make the method more automated. Currently, there are four parameters that must be user-specified: f for edge filtering, a and p_c for keyword extraction, and the minimum size threshold of a major community. We provide some guidelines for setting these parameters in our previous discussion. For example, in setting f we want to filter out a large number of edges while keeping the size of the largest MCC as high as possible. In order for our method to be truly automated, these guidelines need to be formalized so that parameters can be determined without user input.

6.2.2 The Complements Problem

In the complements domain, it is important to more formally characterize the properties of the $comp(c1, c2)$ and $comp_T(c1, c2)$ metrics we define. For example, we normalize for community size by taking percentages of winning bidders rather than absolute numbers, but we need to assess whether the normalization succeeds in practice. One way to answer this question would be to take a larger set of data and see whether size correlates with the complementarity scores for a community. Other metrics can also be developed. For example, it would be useful to have a way of measuring the overall level of complementarity in a market. One way of doing this would be to take an average of the pairwise complementarity

scores. In addition, as in the substitutes case, in order to fully automate the method we would need a way to automatically determine a good value for T. We want T to be low enough that $comp_T(c1, c2)$ can identify complement relationships between many different communities but high enough that $comp_T(c1, c2)$ does not also identify substitute communities as having complement relationships. Finally, as we noted earlier, the ability of the metric to detect complementary relationships between communities depends strongly on whether those communities are well-defined. One way to produce "cleaner" metric scores between communities might be to only look at auctions in each community that contain the top keywords (in terms of highest p_c) for that community. For example, if a community has a camera model as its top keyword, then in evaluating a complementarity score for that community we would look only at auctions that contain that keyword.

There is also room to improve our method for automatically detecting and grouping substitutes communities that have strong complement relationships. In particular, it would be useful to be able to specify the "granularity" at which we want to detect complement relationships. For example, rather than detecting complement relationships between individual models of camera and individual brands and sizes of memory card, we might want to detect more general complement relationships between classes of cameras and types of memory cards. In this case, we would want the method to have fewer, larger supercommunities and fewer, larger subcommunities. Plausible subcommunities would be one for all compact flash cards, one for all secure digital cards, one for cameras that take compact flash memory cards, and one for cameras that take secure digital memory cards. Currently, there is no way to specify to our method the number of communities desired.

6.3 Future Research and Applications

Finally, we turn to avenues for future research. We discuss the possibility of extending our substitutes method to derive a hierarchical categorization of the entire eBay market. We then discuss extending our method to dynamic networks of auctions that are still in progress. Finally, we turn to potential applications of the results in this thesis to other domains with network structure.

6.3.1 Hierarchical Categorization

In this thesis, we examined only the lowest level of the item categorization problem: identifying substitutable goods. In order to be fully applicable to large online auction markets, we would need to generalize our method to higher levels in the hierarchical categorization

of goods. For example, eBay has 33 top-level categories ranging from Antiques to Video Games. Digital Cameras fall in a subcategory of the top-level Cameras and Photo category, while LCD Monitors are nested one level lower, falling in a subcategory of the category Monitors and Projectors, which in turn is a subcategory of the top-level category Computers and Networking. It would be interesting to see the results of applying our method to a larger data set, ideally, one that contains all items on eBay. A hierarchical application of our method where we first apply it to the global data set, then apply it to the individual communities identified, and repeat until we reach a satisfactorily low level, would be one approach to generalizing our method.

One might expect that the lowest level of the categorization problem, which we address, would be the hardest level to solve, since it would seem to require the most specific knowledge about goods, such as knowledge about different models of cameras. However, higher levels of categorization pose their own challenges. Purely keyword-based approaches might fail to give good keywords for higher-level supercategories, since many of the items in these supercategories probably do not contain a good shared keyword. For example, the individual listings in the Consumer Electronics top-level category of eBay probably do not have the phrase "consumer electronics" in their text. Determining the best hierarchy is also non-trivial, even at low levels. For example, in the case of digital cameras, it is not obvious whether it would be better to categorize by resolution and then by brand, or vice versa.

6.3.2 Dynamic Networks

In this thesis, we generated all of our networks from closed auction listings. These listings have the most data about an auction, since we know the closing price and all of the bidder information. Thus, with an application to determining categories for organizing an auction website in mind, it makes sense to use closed auctions because having more auction data makes it easier to generate a better network. Indeed, we make use of the closing price of the auction in our edge filtering method.

Nonetheless, it would be exciting to see whether we can generate useful networks when including auctions that are still in progress. In this dynamic network, structure would change over time as new auctions are added to the system and new cross-auction bids are placed. Characterizing dynamic network structure is an interesting theoretical problem—all of the network analysis literature we have seen in this thesis focuses on static networks and ignores the fact that networks can change over time. The highly temporal nature of auctions on eBay makes it a good data set for developing and empirically evaluating new methods for analyzing dynamic network structure.

As others have noted [22], a disproportionately high fraction of bids are placed near the end time of auctions on eBay. It would be interesting to see whether we could determine meaningful substitutes community information about an auction relatively soon after it is listed, even in the absence of a large amount of current bid data. One way of compensating for the relative sparsity of bid data early in an auction's listing period is to make use of our results from networks constructed on closed listing data. For example, suppose that we have a substitutes community c from running our substitutes method on recently closed listings. Let B_c be the set of all bidders who contributed to the edges in c. We could then look and see whether any bidders in B_c are also bidding in any currently active auctions—if a bidder $b_1 \in B_c$ is a bidder in current auction a_1 and a bidder $b_2 \in B_c$ is a bidder in another current auction a_2, then we could draw an edge between a_1 and a_2. Note that by this definition, we could draw an edge between a_1 and a_2 even if they did not share a common bidder ($b_1 \neq b_2$)—the basis for such edges lies in the fact that they share bidders from a community that has been identified from past data. We could also develop metrics, based on communities derived from past auctions, to assess how directed individual bidders are—that is, how strongly they tend to limit their activity to single communities of goods. In our construction of dynamic networks, we could weight edges from bidders by how directed bidders have been in the past.[1]

Constructing a dynamic network has clear applications for eBay. For example, it would be useful if the site could automatically recategorize items that were incorrectly listed by sellers, while those items were still available for sale. As we saw earlier, our method was able to correctly recategorize at least one such item after it had closed. Better categorization of currently listed items could increase market efficiency, since it would make an item visible to a larger segment of the interested bidder population.

6.3.3 Other Domains and General Lessons

While the specific data set we use comes from the online auction website eBay.com, we expect that the methods used in this thesis would produce similar results when applied to other online auction sites such as uBid.com and Overstock.com. These sites all have readily available bid history information, and as long as our assumptions about bidder behavior hold, we should be able to identify substitutes communities in these markets as well. Our methods could also be extended to other non-auction e-commerce websites where

[1] Some of these ideas, such as considering the directedness of bidders, could be used to potentially improve results for the static substitutes problem as well. After all, the data set of closed listings that we used did not all close at the same time—rather, auctions closed over the entire time period for which data was collected. Thus, we really have a version of the dynamic network structure problem.

hierarchical categorization is important, such as Amazon.com. In this case, we do not have bidder data, since items are sold at a fixed price. However, one could imagine alternate ways of defining edges that also capture substitutability. For example, we could construct an edge between all the items that a user visits in the same web session—if users tend to limit themselves to similar items, then the communities identified on this network could also be useful in defining categories.

As we noted earlier, community detection algorithms have been applied to a wide variety of non-economic domains, including social, biological, and technological networks. While there may not be direct analogues for the substitutes or complements problems in these domains, the results from our work still have relevance. In particular, the finding that edge filtering can result in dramatically improved community divisions suggests that edge filtering might also contribute to improved results in other domains. For example, if one is trying to identify communities of semantically similar web pages, with nodes as web pages and edge defined by hyperlinks, better results might be obtained by identifying and then filtering out "uninformative" links. Simple ways of identifying uninformative links might be considering link placement in the page and the similarity of the link text to the other text in the page. In future work, it would be interesting to examine whether edge filtering can produce improvements in other domains on a similar scale as the improvements we found in the auction domain.

More generally, the dramatic improvements we found from edge filtering touch upon a simple but important point: the results of any network-based approach to solving a problem are constrained by the quality of the network used. Even an "optimal" community detection algorithm—for example, one that finds the global maximum modularity—would produce poor results if the edges in the network it is applied to do not reflect the type of communities desired. It is important to keep this point in mind as network-based approaches are applied to solve problems in a growing set of domains.

Appendix A

Derivation of Update Rules for Greedy Q

In this section, we include a derivation of the update rules for the optimized greedy modularity algorithm. The update rules were first exhibited by Clauset et al. [2], but they were presented without derivation.

We begin by establishing some preliminary properties of ΔQ and then apply these properties to the update rules.

Let

$$q_i = e_{ii} - {a_i}^2 \tag{A.1}$$

Substituting Equation A.1 into Equation 3.2 gives:

$$Q = \sum_i e_{ii} - {a_i}^2 = \sum_i q_i \tag{A.2}$$

Without loss of generality, let us assume we are joining communities a and b into community c. Let us call the set of communities prior to the join P and the set of communities after the join P'. Note that $P' = P + \{c\} - \{a\} - \{b\}$. Then it follows that:

$$\Delta Q_{ab} = \sum_{i \in P'} q_i - \sum_{i \in P} q_i = q_c - [q_a + q_b] \tag{A.3}$$

.

The above equation is true because q_i does not change for any community i that was not involved in the join.

Substituting Equation A.1 into Equation A.3 and simplifying gives:

$$\Delta Q_{ab} = q_c - [q_a + q_b] = e_{ab} - 2a_a a_b \tag{A.4}$$

We now can derive the initialization and update rules.

A.1 Initialization of ΔQ

Initially, every community is exactly one node. Therefore, if node i and node j are connected, $e_{ij} = \frac{1}{M}$; otherwise, $e_{ij} = 0$.[1] Let us then substitute into Equation A.4.

If i and j are connected, we get:

$$\Delta Q_{ij} = e_{ij} - 2a_i a_j = \frac{1}{M} - 2a_i a_j \tag{A.5}$$

Substituting for a_i and a_j gives the initialization Equation 3.5.

If i and j are not connected, we get:

$$\Delta Q_{ij} = e_{ij} - 2a_i a_j = 0 - 2a_i a_j = -2a_i a_j \tag{A.6}$$

Since the algorithm is greedy and we are interested in only the maximum value of Q, we can set $\Delta Q_{ij} = 0$ in the case when ΔQ_{ij} is negative, which gives the initialization Equation 3.6.

A.2 Updating ΔQ

As before, let $\Delta Q'$ be the updated ΔQ matrix and ΔQ be the current matrix. Without loss of generality, let us assume we joined community i and community j into community j'. Let k be one of the other communities.

From Equation A.4, we have $\Delta Q'_{j'k} = e_{j'k} - 2a_{j'}a_k$. We know that $a_{j'} = a_i + a_j$. Substituting, we get:

$$\Delta Q'_{j'k} = e_{j'k} - 2a_i a_k - 2a_j a_k \tag{A.7}$$

.

Now, note that $e_{j'k}$ can be defined in terms of e_{jk} and e_{ik} depending on the connectivity of k and i and j.

If k is connected to both i and j, then $e_{j'k} = e_{ik} + e_{jk}$, so:

$$\Delta Q'_{j'k} = e_{ik} + e_{jk} - 2a_i a_k - 2a_j a_k \tag{A.8}$$

[1] In a slight abuse of notation, here we let i and j refer to both nodes and the communities that those nodes define.

If k is connected to i but not j, then $e_{j'k} = e_{ik}$, so:

$$\Delta Q'_{j'k} = e_{ik} - 2a_i a_k - 2a_j a_k \tag{A.9}$$

If k is connected to j but not i, then $e_{j'k} = e_{jk}$, so:

$$\Delta Q'_{j'k} = e_{jk} - 2a_i a_k - 2a_j a_k \tag{A.10}$$

Finally, if k is connected to neither i nor j, then $e_{j'k} = 0$, so:

$$\Delta Q'_{j'k} = -2a_i a_k - 2a_j a_k \tag{A.11}$$

All that remains is to note that from Equation A.4, we have $\Delta Q_{ik} = e_{ik} - 2a_i a_k$ and $\Delta Q_{jk} = e_{jk} - 2a_j a_k$. If we substitute these terms into our update rules, Equations 3.7, 3.8, and 3.9, we see that they are identical to Equations A.8, A.9, and A.10 as derived above. In the case where k is connected to neither i nor j, $\Delta Q'_{j'k}$ is negative, so we can ignore the update in the greedy implementation.

Appendix B

Comparison of Control to Edge-Filtered Networks

In this appendix, we provide tables comparing basic descriptive statistics for the largest MCCs of control networks to the largest MCCs of their corresponding edge-filtered networks. We present statistics for both price-threshold edge filtering and winning-bidder edge filtering. For each type of filtering, we present statistics for both the Canon and LCD markets.

The data show that the largest MCCs in the price-thresholded control networks are quite similar (in terms of number of edges and nodes) to the largest MCCs in their corresponding price-threshold networks. However, the largest winning-bidder control network MCCs have many more nodes than the MCCs of their corresponding winning-bidder networks.

Network	Edges	Edges in Largest MCC	Nodes in Largest MCC
0.6	22830	22752	3012
Control 0.6	22830	22787(+0.15%)	3032(+0.66%)
0.8	13035	12858	2710
Control 0.8	12858	13004(+1.1%)	2840(+4.8%)

Table B.1: Canon Control Networks vs. Price-Thresholded Networks. The percentage difference (relative to the price-thresholded network) for number of edges in the largest MCC and number of nodes in the largest MCC is given in parentheses.

Network	Edges	Edges in Largest MCC	Nodes in Largest MCC
0.6	60955	60502	5621
Control 0.6	60955	60744(-0.35%)	5458(-2.9%)
0.8	28918	28206	5072
Control 0.8	28918	28743(+1.9%)	4891(-3.6%)

Table B.2: LCD Control Networks vs. Price-Thresholded Networks. The percentage difference (relative to the price-thresholded network) for number of edges in the largest MCC and number of nodes in the largest MCC is given in parentheses.

Network	Edges	Edges in Largest MCC	Nodes in Largest MCC
WBF	12142	11931	2178
Control WBF	12142	12112(+1.5%)	2812(+23%)

Table B.3: Canon Control Networks vs. Winning Bidder Filtering (WBF) Networks. The percentage difference (relative to the WBF network) for number of edges in MCC and number of nodes in MCC is given in parentheses.

Network	Edges	Edges in Largest MCC	Nodes in Largest MCC
WBF	59136	58715	4281
Control WBF	59136	58955(+0.41%)	5474(+27.9%)

Table B.4: LCD Control Networks vs. Winning Bidder Filtering (WBF) Networks. The percentage difference (relative to the WBF network) for number of edges in MCC and number of nodes in MCC is given in parentheses.

Appendix C

The LCD Market: Qualitative Analyses

In this appendix, we include qualitative keyword and community results for the LCD market. We provide the results from the application of our method to three different networks: one with no edge filtering and unweighted edges (Figure C.1), one with price-threshold edge filtering and unweighted edges (Figure C.2), and one with no edge filtering and with weighted edges (Figure C.3).

Similar to Sections 4.2 and 4.3 for the Canon market, we demonstrate that for the LCD market, price-threshold edge filtering results in significantly better community structure and keywords, and that edge weighting results in slightly better community structure and keywords. The results of the LCD market with both edge filtering and edge weighting applied in conjunction are found in Section 4.4 in the main body of the thesis.

C.1 Edge Filtering and Community Keywords

We compared the results of applying our solution to the unweighted LCD network with price thresholding at $f = 0$ (unfiltered) to the results of applying it to the unweighted Canon network with price thresholding at $f = 0.8$. We evaluated the communities and keywords found using qualitative standards for within-community and across-community keywords, as discussed in the next paragraph.

Qualitatively, within a given community, a good set of keywords should be one that defines a category of substitutable goods. In the LCD market, we would likely want keywords to correspond to monitor sizes, brands, and model numbers. We would not want too many

keywords for a community—it would be surprising if many different LCD monitors or models were grouped together in the same community, since these would generally make poor substitutes. Across communities, we would want minimal overlap between keywords, since each community should define a unique set of substitutable goods. In addition, we would want enough communities to be able to identify a range of distinct substitute communities.

For the network with $f = 0$, there were 58 communities ranging in size from 2 to 2486. We define a major community as one whose size is at least 1% of the total number of auctions in the largest MCC. In the network with $f = 0$, there were 4 such communities. We applied keyword extraction with $a = 0.01$ and $p = 0.3$ and examined the keywords for those 4 communities. Of these, 3 communities had significant keywords, with a range of 3 to 5 keywords each. The major communities with significant keywords are depicted in Figure C.1. There were 2 monitor sizes (17", 19") and 1 monitor model (2005fpw) in the set of significant keywords. No community had more than 1 monitor size or monitor model keyword. Across communities, there was no overlap between LCD size or model keywords.

For the network with $f = 0.8$, there were 66 communities ranging in size from 3 to 1174. Of these, 10 were major communities. We applied keyword extraction with the same parameters and examined the keywords. Eight of the major communities had significant keywords, with a range of 1 to 6 significant keywords per community. The major communities with significant keywords are depicted in Figure C.2. There were 4 monitor sizes (15", 17", 19", 24") and 2 monitor models (2005fpw, 2405fpw) in the set of significant keywords. Again, no community had more than 1 monitor size or monitor model keyword, and across communities, there was no overlap between camera model keywords. There appears to be some over-segmentation, since, for example, there are 4 different communities that have 17" as a keyword.

The keywords and communities from the $f = 0.8$ network are qualitatively better. In particular, the $f = 0.8$ network identifies a larger set of monitor sizes and monitor models. While there is some over-segmentation in the $f = 0.8$ network, this is arguably preferable to completely missing two sizes of monitors in the market (15" and 24") and one model (2405fpw).

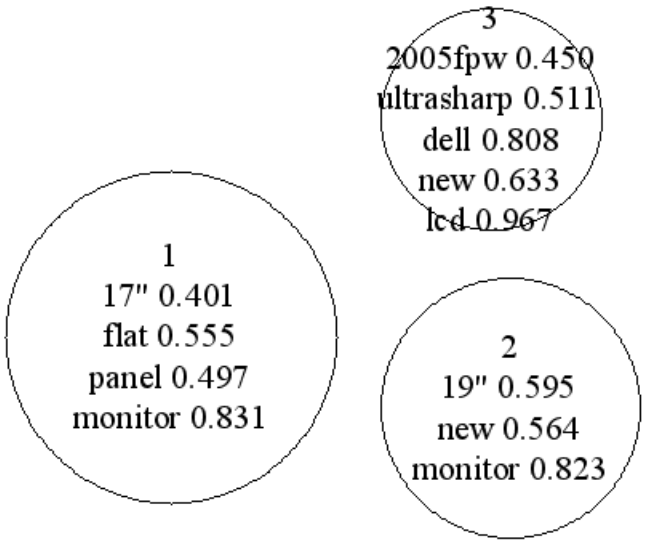


Figure C.1: LCD Communities with No Edge Filtering and with Unweighted Edges. The size of each circle is proportional to the number of nodes in each community. Significant keywords ($a \leq 0.01$, $p_c \geq 0.3$) and their associated p_c are listed in order of descending a (most overrepresented keyword first). Since the a values are all small, they are not shown. Placement of the communities in the figure is arbitrary.

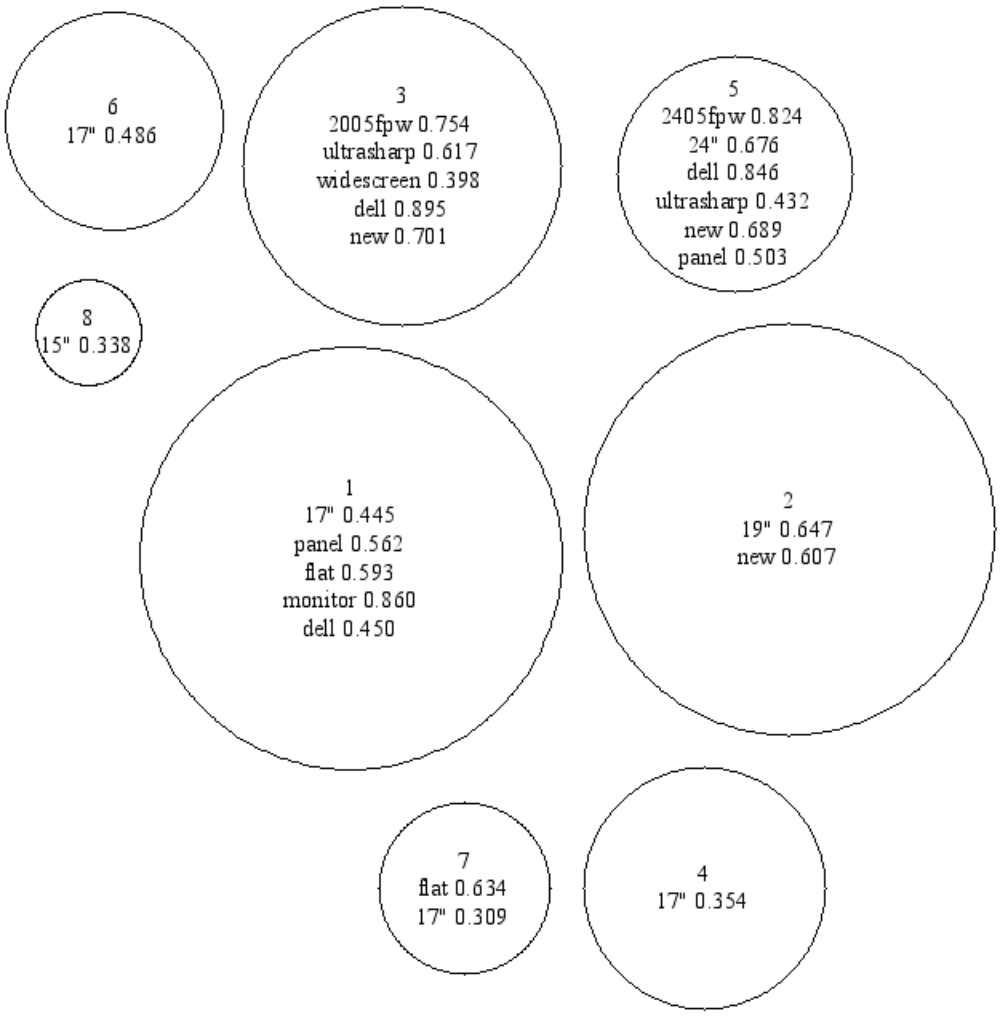


Figure C.2: LCD Communities with 0.8 Edge Thresholding and with Unweighted Edges. The size of each circle is proportional to the number of nodes in each community. Significant keywords ($a \leq 0.01$, $p_c \geq 0.3$) and their associated p_c are listed in order of descending a (most overrepresented keyword first). Since the a values are all small, they are not shown. Placement of the communities in the figure is arbitrary.

C.2 Edge Weighting and Community Keywords

We applied our solution to an un-edge-filtered, weighted LCD network and compared the results to those for the un-edge-filtered, unweighted network in the previous section (see Figure C.1).

For the weighted network, there were 28 communities ranging in size for 3 to 2089. Of these, 6 were major communities. We applied keyword extraction with the same parameters and examined the keywords. All 6 of the major communities had significant keywords, with a range of 1 to 7 significant keywords per community. These major communities are shown in Figure C.3. There were two monitor sizes (17", 19") and two monitor models (e173fp, 2005fpw) in the set of significant keywords. In addition, one community had a keyword for projectors (projector). Again, no community had more than 1 monitor size or monitor model keyword, and across communities, there was no overlap between camera model keywords.

The performance is similar to that of the unweighted LCD network. However, one could argue that it is slightly better because it identifies another model type (e173fp) as well as a category of non-monitors that were in the data (projectors).

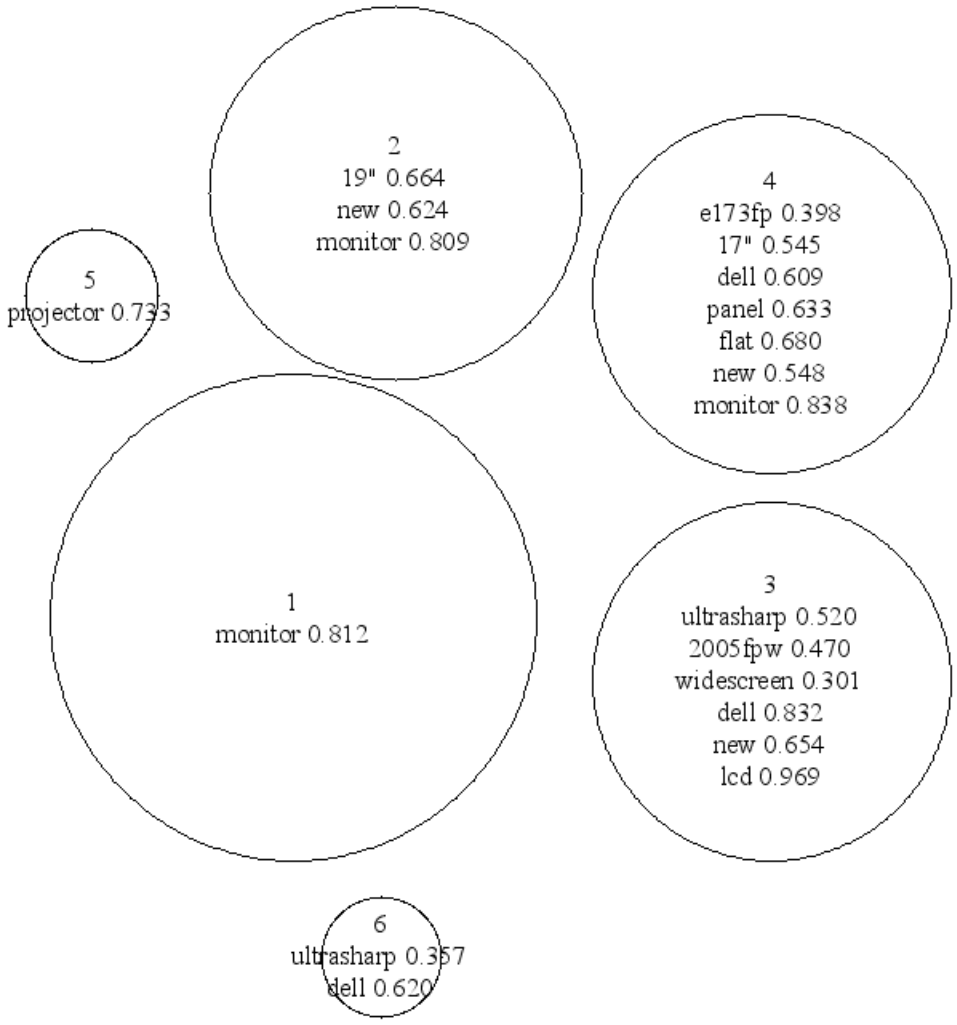


Figure C.3: LCD Communities with No Edge Filtering and with Weighted Edges. The size of each circle is proportional to the number of nodes in each community. Significant keywords ($a \leq 0.01$, $p_c \geq 0.3$) and their associated p_c are listed in order of descending a (most overrepresented keyword first). Since the a values are all small, they are not shown. Placement of the communities in the figure is arbitrary.

Appendix D

Keyword Lists

In this section, we give lists of keywords. The first two tables contain lists of keywords for the global Canon set and the global LCD set. All keywords present in at least 1% of auctions are listed.

Then, we give lists of keywords for major communities of the final Canon and LCD networks analyzed in this paper: the price threshold 0.8, weighted networks. We list all keywords that are present in at least 5% of the auctions in their community.

canon 0.994	a410 0.031	a510 0.014
powershot 0.930	s500 0.028	a75 0.014
digital 0.665	w/ 0.027	mb 0.014
camera 0.487	warranty 0.026	g3 0.014
megapixel 0.417	5mp 0.025	12x 0.014
new 0.324	s50 0.025	reserve 0.014
elph 0.272	sd200 0.024	&amp 0.014
5.0 0.185	s410 0.024	128mb 0.013
3.2 0.126	s1 0.024	a95 0.013
7.1 0.125	512mb 0.022	4x 0.013
mp 0.125	extras 0.021	used 0.013
4.0 0.101	2.1 0.020	g6 0.013
is 0.099	5.0mp 0.020	sd-550 0.013
a620 0.093	shot 0.019	g2 0.012
brand 0.079	box 0.019	case 0.012
s2 0.067	2.0 0.019	a400 0.012
usa 0.066	256mb 0.019	cf 0.011
sd400 0.065	plus 0.018	opt 0.011
sd500 0.059	7.1mp 0.018	memory 0.011
sd450 0.048	1 0.018	s30 0.011
a610 0.048	power 0.018	s1is 0.011
sd 0.046	4.0mp 0.017	s230 0.011
kit 0.044	s80 0.017	500 0.011
a70 0.042	sd300 0.016	for 0.011
3.2mp 0.041	a85 0.016	s110 0.011
a520 0.041	like 0.016	pro1 0.011
1gb 0.038	no 0.015	nib 0.010
nr 0.037	with 0.015	w/1gb 0.010
card 0.035	4mp 0.015	s200 0.010
sd550 0.034	sd-500 0.015	as 0.010
zoom 0.034	3x 0.014	
s400 0.031	mega 0.014	

Table D.1: Keywords for the Global Canon Set with the Fraction of Auctions Matching. All 94 keywords matching 1% or more of auctions are listed. In total, there were 899 keywords.

lcd 0.952
monitor 0.805
flat 0.493
new 0.478
panel 0.441
dell 0.415
17” 0.228
19” 0.211
ultrasharp 0.150
brand 0.136
15” 0.121
2005fpw 0.106
e173fp 0.095
samsung 0.085
sony 0.085
tft 0.079
screen 0.074
widescreen 0.069
black 0.063
2405fpw 0.063
inch 0.056
24” 0.051
20” 0.051
syncmaster 0.050
box 0.048
nr 0.048
computer 0.047
hp 0.046
17 0.044
color 0.041
viewsonic 0.040
sealed 0.040
19 0.039
914v 0.038
20.1 0.037
mag 0.033
sdm-hs95/b 0.032
nec 0.031
wide 0.031
nib 0.030
e193fp 0.030
display 0.030
in 0.029
like 0.027
20.1” 0.025
18” 0.023
monitor, 0.022
15 0.020
acer 0.020
2405 0.020
sdm-hs95 0.019
1905fp 0.018
warranty 0.018
dvi 0.016
no 0.016
1800fp 0.016
model 0.015
reserve 0.015
digital 0.014
speakers 0.014
18 0.014
innovision 0.013
flat-panel 0.013
w/ 0.012
gateway 0.012
multisync 0.012
pc 0.012
princeton 0.012
new 0.012
ibm 0.011
with 0.011
pavilion 0.011
tv 0.011
e153fp 0.011
kds 0.011
lt916s 0.010
emachines 0.010
envision 0.010

Table D.2: Keywords for the Global LCD Set with the Fraction of Auctions Matching. All 78 keywords matching 1% or more of auctions are listed. In total, there were 2535 keywords.

Table D.3: Major Communities and All Keywords of $p_c > 0.05$ for the Canon Network with Price Thresholding $f = 0.8$ and Edge Weights. For each community, keywords are sorted (left-right, top-bottom) in ascending order of a (the most overrepresented keywords are listed first.) The format for each entry is *keyword* a p_c. Communities with significant keywords have, in parentheses, the number of the corresponding community in Figure 4.6.

Community 2077 (1) of size 339		
sd400 0.000 0.451	sd450 0.000 0.322	5.0 0.000 0.575
elph 0.000 0.676	digital 0.000 0.894	new 0.000 0.528
w/1gb 0.000 0.050	5mp 0.000 0.074	s500 0.000 0.068
card 0.000 0.080	megapixel 0.020 0.472	powershot 0.435 0.932
brand 0.648 0.074	mp 0.884 0.103	canon 0.989 0.985
Community 1200 (2) of size 329		
sd500 0.000 0.432	sd550 0.000 0.258	7.1 0.000 0.480
sd-500 0.000 0.116	sd-550 0.000 0.097	elph 0.000 0.581
500 0.000 0.082	digital 0.000 0.851	sd 0.000 0.125
7.1mp 0.000 0.055	new 0.000 0.438	brand 0.022 0.109
kit 0.073 0.061	powershot 0.137 0.945	megapixel 0.335 0.429
usa 0.487 0.067	camera 0.926 0.447	canon 0.991 0.985
mp 0.996 0.076		
Community 174 of size 287		
s1 0.000 0.157	repair 0.000 0.084	s1is 0.000 0.094
a95 0.000 0.101	cameras 0.000 0.084	parts 0.000 0.063
a70 0.000 0.122	is 0.000 0.202	3.2 0.000 0.237
sd300 0.000 0.056	2.1 0.000 0.056	3.2mp 0.003 0.073
canon 0.103 1.000	camera 0.305 0.502	4.0 0.415 0.105
mp 0.485 0.125	megapixel 0.899 0.380	powershot 0.913 0.909
digital 0.959 0.617	elph 0.994 0.206	5.0 0.998 0.118
Community 2363 (3) of size 276		
a620 0.000 0.833	new 0.000 0.844	7.1 0.000 0.475
kit 0.000 0.192	620 0.000 0.062	usa 0.000 0.239
a610 0.000 0.163	7.1mp 0.000 0.087	512mb 0.000 0.091
camera 0.000 0.714	256mb 0.000 0.080	4x 0.000 0.062
not 0.000 0.051	brand 0.000 0.181	1gb 0.000 0.091

Table D.3: Major Communities and All Keywords of $p_c > 0.05$ for the Canon Network with Price Thresholding $f = 0.8$ and Edge Weights. (continued).

mp 0.011 0.170	powershot 0.042 0.957	canon 0.108 1.000
sd 0.268 0.054	digital 0.430 0.670	megapixel 0.998 0.330
Community 2072 of size 275		
a410 0.000 0.193	a610 0.000 0.236	a400 0.000 0.069
a510 0.000 0.062	s400 0.000 0.084	3.2mp 0.000 0.102
3.2 0.000 0.222	mp 0.010 0.171	camera 0.010 0.556
canon 0.108 1.000	new 0.312 0.338	4.0 0.323 0.109
powershot 0.567 0.927	5.0 0.774 0.167	digital 0.846 0.636
brand 0.859 0.062	megapixel 0.999 0.327	elph 1.000 0.164
Community 408 (4) of size 204		
s2 0.000 0.848	is 0.000 0.853	12x 0.000 0.181
opt 0.000 0.137	mem.crd 0.000 0.127	5.0 0.000 0.564
5.0mp 0.000 0.152	1gb 0.000 0.206	s2is 0.000 0.074
plus 0.000 0.127	zoom 0.000 0.162	sd 0.000 0.176
brand 0.000 0.206	new 0.000 0.436	megapixel 0.102 0.461
kit 0.252 0.054	canon 0.452 0.995	mp 0.453 0.127
usa 0.763 0.054	camera 0.957 0.426	
Community 69 (5) of size 153		
a70 0.000 0.412	a85 0.000 0.196	yr 0.000 0.137
3.2 0.000 0.438	1 0.000 0.137	warranty 0.000 0.144
3.2mp 0.000 0.183	s230 0.000 0.072	usa 0.000 0.137
mp 0.001 0.209	camera 0.001 0.614	digital 0.018 0.745
powershot 0.035 0.967	canon 0.178 1.000	4.0 0.437 0.105
megapixel 0.992 0.320	elph 0.998 0.170	
Community 2163 (6) of size 150		
a520 0.000 0.460	4.0 0.000 0.400	sealed 0.000 0.053
s1 0.000 0.093	a510 0.000 0.067	a410 0.000 0.100
3.2 0.000 0.220	brand 0.001 0.147	megapixel 0.005 0.520
new 0.545 0.320	canon 0.574 0.993	is 0.584 0.093
powershot 0.786 0.913	camera 0.905 0.433	mp 0.951 0.080
Community 2508 of size 131		
g2 0.000 0.214	s200 0.000 0.122	a300 0.000 0.115

Table D.3: Major Communities and All Keywords of $p_c > 0.05$ for the Canon Network with Price Thresholding $f = 0.8$ and Edge Weights. (continued).

g1 0.000 0.076	2.0 0.000 0.145	3.3 0.000 0.061
a40 0.000 0.076	s230 0.000 0.076	extras 0.000 0.092
w/ 0.000 0.099	3.2 0.000 0.229	4.0 0.002 0.176
megapixel 0.014 0.511	powershot 0.138 0.954	canon 0.197 1.000
mp 0.921 0.084	camera 0.937 0.420	elph 0.996 0.168
digital 0.999 0.542		
Community 185 (7) of size 119		
s50 0.000 0.420	s30 0.000 0.143	s45 0.000 0.067
n-sony 0.000 0.059	5 0.000 0.067	4.0mp 0.000 0.076
4mp 0.000 0.059	used 0.000 0.050	3.2 0.007 0.202
s400 0.010 0.067	extras 0.014 0.050	5.0 0.017 0.261
mp 0.190 0.151	powershot 0.200 0.950	canon 0.208 1.000
megapixel 0.399 0.429	4.0 0.618 0.092	camera 0.638 0.471
Community 645 (8) of size 115		
sd200 0.000 0.435	sd-200 0.000 0.217	g3 0.000 0.165
sd300 0.000 0.104	4.0 0.000 0.278	elph 0.000 0.452
digital 0.000 0.843	3.2 0.001 0.226	nr 0.192 0.052
canon 0.212 1.000	megapixel 0.222 0.452	powershot 0.226 0.948
camera 0.424 0.496	mp 0.934 0.078	5.0 1.000 0.052
Community 2195 (9) of size 69		
pro1 0.000 0.377	g6 0.000 0.406	8 0.000 0.319
pro 0.000 0.116	year 0.000 0.072	1 0.000 0.174
7.1 0.000 0.333	with 0.002 0.058	like 0.003 0.058
megapixel 0.006 0.565	warranty 0.008 0.072	usa 0.016 0.130
powershot 0.090 0.971	nr 0.177 0.058	canon 0.268 1.000
mp 0.587 0.116	camera 0.911 0.406	new 1.000 0.087
Community 1935 (10) of size 61		
s80 0.000 0.639	8.0 0.000 0.164	model 0.000 0.131
8.0-megapixel 0.000 0.082	usa 0.000 0.246	brand 0.000 0.262
a520 0.000 0.148	new 0.000 0.541	kit 0.000 0.131
1gb 0.007 0.098	camera 0.031 0.607	4.0 0.215 0.131
powershot 0.261 0.951	canon 0.874 0.984	digital 0.893 0.590

Table D.3: Major Communities and All Keywords of $p_c > 0.05$ for the Canon Network with Price Thresholding $f = 0.8$ and Edge Weights. (continued).

mp 0.919 0.066	3.2 0.923 0.066	megapixel 0.953 0.311
elph 1.000 0.082		
Community 2636 (11) of size 56		
s410 0.000 0.446	*new-demo* 0.000 0.107	s-410 0.000 0.054
4.0 0.000 0.518	s400 0.000 0.250	elph 0.000 0.750
s500 0.000 0.161	sd110 0.000 0.071	4.0mp 0.001 0.071
megapixel 0.104 0.500	mp 0.111 0.179	card 0.231 0.054
powershot 0.314 0.946	digital 0.417 0.679	5.0 0.791 0.143
canon 0.893 0.982	3.2 0.949 0.054	camera 1.000 0.214
new 1.000 0.054		
Community 2497 (12) of size 31		
s110 0.000 0.484	2.1 0.000 0.516	s100 0.000 0.161
nice 0.000 0.065	elph 0.000 0.710	2.0 0.001 0.097
nr 0.003 0.129	digital 0.020 0.839	megapixel 0.131 0.516
canon 0.339 1.000	powershot 0.451 0.935	mp 0.681 0.097
3.2 0.849 0.065	camera 0.998 0.226	

Table D.4: Major Communities and All Keywords of $p_c > 0.05$ for the LCD Network with Price Thresholding $f = 0.8$ and Edge Weights. For each community, keywords are sorted (left-right, top-bottom) in ascending order of a (the most overrepresented keywords are listed first.) The format for each entry is *keyword* a p_c. Communities with significant keywords have, in parentheses, the number of the corresponding community in Figure 4.7.

Community 4618 (1) of size 973		
19” 0.000 0.692	sony 0.000 0.304	914v 0.000 0.177
sdm-hs95/b 0.000 0.144	samsung 0.000 0.230	sdm-hs95 0.000 0.089
19 0.000 0.124	syncmaster 0.000 0.133	black 0.000 0.149
mag 0.000 0.087	new 0.000 0.629	brand 0.000 0.206
tft 0.000 0.127	sealed 0.000 0.075	nib 0.000 0.061
e193fp 0.000 0.061	box 0.018 0.063	monitor 0.503 0.805

Table D.4: Major Communities and All Keywords of $p_c > 0.05$ for the LCD Network with Price Thresholding $f = 0.8$ and Edge Weights. (continued).

inch 0.695 0.052	lcd 1.000 0.928	
Community 1200 (2) of size 754		
e173fp 0.000 0.438	17” 0.000 0.638	panel 0.000 0.650
17 0.000 0.122	flat 0.000 0.668	dell 0.000 0.581
brand 0.000 0.239	new 0.000 0.611	color 0.000 0.090
like 0.000 0.052	hp 0.001 0.070	box 0.001 0.073
monitor 0.005 0.842	sealed 0.032 0.053	lcd 0.046 0.966
tft 0.680 0.074	inch 0.703 0.052	screen 0.987 0.053
Community 4075 (3) of size 637		
15” 0.000 0.435	15 0.000 0.069	nec 0.000 0.085
viewsonic 0.000 0.068	display 0.000 0.052	tft 0.007 0.105
hp 0.052 0.060	screen 0.072 0.089	monitor 0.175 0.819
panel 0.396 0.446	black 0.593 0.061	flat 0.605 0.488
lcd 0.811 0.945	samsung 0.997 0.055	17” 0.998 0.181
Community 2077 (4) of size 583		
2005fpw 0.000 0.839	widescreen 0.000 0.441	ultrasharp 0.000 0.640
20.1 0.000 0.281	20” 0.000 0.317	dell 0.000 0.916
20.1” 0.000 0.161	wide 0.000 0.136	new 0.000 0.714
sealed 0.050 0.053	box 0.093 0.060	lcd 0.134 0.962
screen 0.906 0.060	brand 0.974 0.108	
Community 666 (5) of size 446		
1905fp 0.000 0.159	viewsonic 0.000 0.094	hp 0.000 0.103
15” 0.000 0.197	19” 0.000 0.296	like 0.000 0.054
syncmaster 0.002 0.081	monitor 0.002 0.859	17” 0.008 0.276
samsung 0.020 0.112	lcd 0.084 0.966	ultrasharp 0.117 0.170
nr 0.280 0.054	flat 0.579 0.489	screen 0.582 0.072
inch 0.664 0.052	panel 0.667 0.430	tft 0.946 0.058
dell 0.986 0.363		
Community 2363 (6) of size 391		
24” 0.000 0.660	2405fpw 0.000 0.803	2405 0.000 0.251
dell 0.000 0.859	ultrasharp 0.000 0.453	wide 0.000 0.174
24 0.000 0.064	new 0.000 0.696	widescreen 0.000 0.179

Table D.4: Major Communities and All Keywords of $p_c >$ 0.05 for the LCD Network with Price Thresholding $f = 0.8$ and Edge Weights. (continued).

panel 0.014 0.496	lcd 0.036 0.972	brand 0.071 0.161
nr 0.220 0.056	inch 0.253 0.064	flat 0.341 0.504
screen 0.723 0.066	monitor 0.836 0.785	2005fpw 0.986 0.072
Community 2195 (7) of size 278		
e173fp 0.000 0.331	17” 0.000 0.536	color 0.000 0.115
computer 0.000 0.126	17 0.000 0.097	flat 0.000 0.615
inch 0.000 0.104	panel 0.001 0.536	screen 0.001 0.122
dell 0.021 0.475	15” 0.029 0.158	nr 0.030 0.072
viewsonic 0.040 0.061	lcd 0.040 0.975	box 0.060 0.068
monitor 0.081 0.838	brand 0.618 0.129	tft 0.742 0.068
new 0.952 0.428		
Community 3017 (8) of size 267		
18 0.000 0.184	1800fp 0.000 0.187	18” 0.000 0.217
computer 0.000 0.258	e193fp 0.000 0.165	flat 0.000 0.719
dell 0.000 0.521	screen 0.000 0.127	monitor 0.002 0.876
19” 0.002 0.281	ultrasharp 0.014 0.199	17 0.017 0.071
color 0.099 0.056	black 0.155 0.079	lcd 0.220 0.963
brand 0.250 0.150	panel 0.340 0.453	inch 0.500 0.056
tft 0.595 0.075	17” 0.761 0.210	e173fp 0.871 0.075
sony 0.890 0.064	new 0.984 0.412	
Community 2673 of size 189		
parts 0.000 0.143	repair 0.000 0.095	for 0.000 0.106
or 0.000 0.053	15” 0.000 0.275	model 0.000 0.063
viewsonic 0.000 0.095	15 0.001 0.053	screen 0.003 0.127
hp 0.006 0.085	display 0.010 0.058	nr 0.021 0.079
inch 0.083 0.079	color 0.196 0.053	17 0.284 0.053
tft 0.696 0.069	17” 0.703 0.212	lcd 0.980 0.921
flat 1.000 0.339		
Community 906 of size 99		
lg 0.000 0.071	nec 0.000 0.121	multisync 0.000 0.051
pc 0.000 0.051	acer 0.002 0.061	sony 0.021 0.141
17” 0.022 0.313	tft 0.059 0.121	like 0.072 0.051

Table D.4: Major Communities and All Keywords of $p_c >$ 0.05 for the LCD Network with Price Thresholding $f = 0.8$ and Edge Weights. (continued).

display 0.109 0.051	hp 0.122 0.071	19 0.134 0.061
color 0.157 0.061	syncmaster 0.173 0.071	inch 0.265 0.071
samsung 0.284 0.101	19” 0.298 0.232	monitor 0.369 0.818
lcd 0.370 0.960	black 0.384 0.071	computer 0.437 0.051
brand 0.550 0.131	15” 0.731 0.101	new 1.000 0.303
flat 1.000 0.293	panel 1.000 0.232	
Community 3166 (9) of size 95		
1704fpt 0.000 0.179	90 0.000 0.084	1704fp 0.000 0.053
day 0.000 0.095	refurbished 0.000 0.084	computer 0.000 0.242
warranty 0.000 0.084	1800fp 0.000 0.074	17” 0.000 0.379
tft 0.002 0.158	monitor 0.003 0.916	nib 0.030 0.063
18” 0.030 0.053	sealed 0.122 0.063	screen 0.125 0.105
inch 0.230 0.074	nr 0.243 0.063	box 0.250 0.063
17 0.348 0.053	hp 0.381 0.053	19” 0.401 0.221
dell 0.449 0.421	ultrasharp 0.531 0.147	samsung 0.777 0.063
panel 0.887 0.379	15” 0.922 0.074	lcd 0.954 0.916
brand 0.961 0.074	flat 0.966 0.400	new 1.000 0.284
Community 4533 (10) of size 88		
/new 0.000 0.091	multi 0.000 0.091	inches 0.000 0.091
sync 0.000 0.091	7004201 0.000 0.068	fpd1830 0.000 0.068
gateway 0.000 0.102	e15t4 0.000 0.057	nec 0.000 0.148
15” 0.000 0.330	in 0.000 0.125	model 0.000 0.080
2001fp 0.000 0.057	emachines 0.000 0.057	box 0.000 0.125
19 0.006 0.091	monitor 0.027 0.886	like 0.041 0.057
nib 0.072 0.057	viewsonic 0.093 0.068	lcd 0.137 0.977
screen 0.159 0.102	tft 0.208 0.102	flat 0.223 0.534
sony 0.275 0.102	hp 0.316 0.057	panel 0.396 0.455
inch 0.490 0.057	black 0.601 0.057	ultrasharp 0.968 0.080
17” 0.989 0.125		

Acknowledgements

This thesis would not have been possible without the help I received from my wonderful advisors, Prof. David Parkes and Prof. Patrick Wolfe. They were always a source of ideas and encouragement. I would also like to thank Jennifer Caswell for editing the thesis. Finally, I would like to thank my parents for their support.

Bibliography

[1] A. Barabási, R. Albert, and H. Jeong. Mean-field theory for scale-free random networks. *Physica A*, 272:173–189, 1999.

[2] A. Clauset, M. Newman, and C. Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, 2004.

[3] L. Danon, J. Duch, A. Diaz-Guilera, and A. Arenas. Comparing community structure identification. *J. Stat. Mech.*, page P09008, 2005.

[4] eBay. eBay announces fourth quarter and full year 2005 financial results, 2005.

[5] G. Flake, S. Lawrence, C. Giles, and F. Coetzee. Self-organization and identification of web communities. *IEEE Computer*, 35:66–71, 2002.

[6] M. Girvan and M. Newman. Community structure in social and biological networks. *PNAS*, 99:7821–7826, 2002.

[7] J. Hahn. The dynamics of mass online marketplaces: A case study of an online auction. In *Proc. ACM SIGCHI on Human Factors in Computing Systems*, pages 317–324, 2001.

[8] RKX. Jin, A. Sanghvi, and M. Zuckerberg. Developing arbitrage strategies for online auctions. *Working Paper*, 2004.

[9] B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. In *Proc. ACM Conference on EC*, pages 18–28, 2001.

[10] G. Lohse and P. Spiller. Electronic shopping: The effect of customer interfaces on traffic and sales. *Comm. of the ACM*, 41:81–87, 1998.

[11] D. McGuinness. Ontologies and online commerce. *IEEE Intelligent Systems*, 16:9–10, 2001.

[12] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized on-line matching. In *Proc. IEEE FOCS*, 2005.

[13] M. Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 1:226–251, 2004.

[14] M. Newman. Mixing patterns in networks. *Phys. Rev. E*, 67:026126, 2003.

[15] M. Newman. Analysis of weighted networks. *Phys. Rev. E*, 70:056131, 2004.

[16] M. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69:066133, 2004.

[17] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[18] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of networks in nature and society. *Nature*, 435:814–818, 2005.

[19] J. Reichardt and S. Bornholdt. Detecting fuzzy community structures in complex networks with a Potts model. *Phys. Rev. Lett.*, 93:218701, 2004.

[20] J. Reichardt and S. Bornholdt. Economic networks and social communities in online-auction sites. *Preprint*, 2006.

[21] A. Roth, A. Juda, and D. Parkes. Unsupervised learning of bidding strategies for sequential auctions. *Working Paper*, 2005.

[22] A. Roth and A. Ockenfels. Last minute bidding and the rules for ending second-price auctions: Evidence from eBay and Amazon auctions on the Internet. *American Economic Review*, 92:1093–1103, 2002.

[23] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513–523, 1988.

[24] H. Shah, N. Joshi, A. Sureka, and P. Wurman. Mining for bidding strategies on eBay. In *Lecture Notes on Articial Intelligence*. Springer-Verlag, 2003.

[25] L. Weiss, M. Capozzi, and L. Prusak. Learning from the Internet giants. *MIT Sloan Management Review*, 45:79–84, 2004.

[26] I. Yang, H. Jeong, B. Kahng, and A.L. Barabási. Emerging behavior in electronic bidding. *Phys. Rev. E*, 68:016102, 2003.