University of Cincinnati

Date: 6/25/2015

I, Feng Mai , hereby submit this original work as part of the requirements for the degree of Doctor of Philosophy in Business Administration.

It is entitled:

Essays in Business Analytics

Student's name: **Feng Mai**

This work and its defense approved by:

Committee chair: Michael Fry, Ph.D.

Committee member: Jeffrey W. Ohlmann, Ph.D.

Committee member: Hsiang-Li Chiang, Ph.D.

Committee member: David Curry, Ph.D.

16192

Essays in Business Analytics

A dissertation submitted to the
Graduate School
of the University of Cincinnati
in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

in the Department of Operations, Business Analytics, and Information Systems
of the Carl H. Lindner College of Business
by

Feng Mai

M.S. Statistics, Miami University, 2010
M.A. Mathematics, Wabash College, 2008

June 2015

Committee Chair: Michael J. Fry, Ph.D.

Abstract

The availability of structured and unstructured data, along with recent advancements in machine learning methods and tools, pose both challenges and opportunities for businesses. The three essays in this dissertation address important aspects of business such as marketing and operations using emerging business analytics methods. The essays are devoted to two topics in analytics: advances in unsupervised learning methods and analytics of unstructured, textual data.

In Essay 1 we develop a business intelligence framework and advance market structure analysis by combining computational linguistics, machine learning, and relevant marketing theories to reveal consumer insights from free-form product reviews. Our text analytics method is able to create a hierarchy for product attributes, discover consumer sentiments, and construct market structure perceptual maps. In Essay 2, we use deep learning and evolutionary clustering to study the dynamics of market segmentation. We adopt the skip-gram model to learn computable, vectorized representation of product attributes. In addition, the evolutionary clustering model integrates a measure of temporal smoothness into the overall measure of clustering quality, and thus can be used as a method to study market structures over time. In Essay 3, we apply expectation-maximization (EM), a widely used method in statistical inference, to solve a discrete optimization problem that has many applications in operations management. We frame the optimization problem as a semi-supervised learning problem and develop a heuristic to solve a capacitated clustering problem and its stochastic variant.

Acknowledgements

First and foremost, I would like to express the deepest appreciation to my advisor, Mike Fry, and committee members Roger Chiang, David Curry, and Jeff Ohlmann. They provided invaluable guidance and generous support. Their dedication towards research has been and will always be motivational for me. I also wish to thank the professors in the OBAIS department. I am sincerely grateful for what I have learnt from them.

I appreciate my mother, Zhengming Lin, for her love throughout the years. I thank my friends and fellow PhD students, especially Xin, Wei, Muer, SK, CJ, for their encouragement and support.

Finally, I dedicate this dissertation to the loving memory of Han Jiang.

Table of Contents

1 Mining Consumer-Generated Product Reviews to Automate Market Structure Analyses

1.1 INTRODUCTION

The first decade of twenty-first century is the era of business intelligence and social media. The increasing popularity of social media has resulted in astounding growth in the amount of digital data available, up to an approximated 2.7 zettabytes (i.e., 2.7×10^{21} bytes) in 2012 (IDC 2011), 80% of which are unstructured. Crowdsourcing systems have prompted great growth in user-generated content (UGC), in terms of both volume and significance (Doan et al. 2011) and in the form of product reviews, blogs, and other consumer-initiated contributions (Fader and Winer 2012). Consumer-initiated marketing activities established by leading vendors such as Amazon, eBay, and Netflix also significantly enhance this big data phenomenon. Online product reviews contributed freely by consumers and readily available could be a valuable information source for marketing research. Unlike transaction records collected from legacy systems and consumers' experience and opinions, obtained through consumer surveys and

interviews, online product reviews contain a large volume and rich consumer insights and behavioral information. Academics and practitioners have started tapping this new data source, in an attempt to listen better to the voice of the customer (VOC) expressed in reviews.

Elrod and Keane (1995) define a market structure as a means to explain consumer preferences in terms of product attributes. Market structure analysis identifies the extent to which different brands compete in the marketplace (Hansen and Singh 2009). Traditional market structure analysis methods analyze survey and interview results to define substitution and complementary relationships among competing brands. The market structure is largely determined by customers (Elrod et al. 2002), and is influenced by their particular usage situations and experiences (Ratneshwar et al. 1999).

The use of UGC for market structure analysis has become an emerging, important marketing research topic. As Lee and Bradlow describe (2011, p. 882), "the preponderance of opinion, as represented in the continuous stream of reviews over time, provides practical input to augment traditional approaches (e.g., surveys, focus groups) for conducting brand sentiment analysis and can be done (unlike traditional methods) continuously, automatically, inexpensively, and in real time." According to Netzer et al. (2012, p. 521), by analyzing online product reviews, "firms could, in principle, gain a better understanding of the online discussion and the marketing opportunities, the market structure, the competitive landscape, and the features of their own and their competitors' product that consumers discuss." However, a challenge remains, due to the sheer volume, unstructured nature, and lack of tools available to explore UGC effectively. Text analytics techniques and the related field of big data analytics offer increasingly prominent ways to tackle the challenge (Chen et al. 2012; LaValle et al. 2011).

The objective of the current research is to advance market structure analyses by combining marketing theories and advances in computational linguistics to reveal deeper market insights from product reviews. We develop an innovative text analytics framework to create a hierarchy for product attributes, discover consumer sentiments, and construct market structure perceptual maps. Figure 1 presents our research premise and positions the text analytics approach in comparison with a traditional market structure analysis approach. In the text analytics approach, we treat consumers who contribute reviews as a giant focus group. We explore and analyze online product reviews to capture VOC, without asking consumers a single question.

Insert Figure 1 about here

Pioneering work by Lee and Bradlow (2011) and Netzer et al. (2012) establishes the foundation of a text analytics approach for conducting market structure analysis using UGC. Our research enhances this emerging approach by filling two gaps. First, whereas researchers have conducted market structure analysis using various analysis techniques, they tend to make only limited use of textual information, such as pro–con lists (Decker and Trusov 2010; Lee and Bradlow 2011). Free-form reviews arguably are more common forms of UGC but largely ignored as data sources for marketing research. As shown in Figure 2, information about how consumers use the products and their opinions of specific product attributes appear in most free-form reviews. We demonstrate how text-mining methods can distill the information central to market structure analysis while unveiling additional knowledge too.

Second, traditional market structure analysis starts by providing "a summary of customer perceptions and evaluations of existing products in terms of product attributes" (Elrod et al. 2002, p. 223), which it uses to explain market competition. This aspect of interpretability is lacking in Lee and Bradlow's (2011) and Netzer et al.'s (2012) approaches, which cannot

interpret brands' similarities without running additional analyses. With sentiment analysis using machine learning methods, we seek to establish and validate a more principled way to automate market structure analyses. Our framework incorporates the valuation of product attributes into the process for establishing perceptual similarities among brands.

Insert Figure 2 about here

In the next section, we review literature and relevant theories and discuss our contributions. We then introduce our proposed method to obtain market structures. As a proof-of-concept evaluation, we conduct an empirical study to analyze more than 45,000 consumer-generated reviews of tablet computers and demonstrate the validity of our framework. Finally, we conclude with marketing implications and research directions. The Web Appendix contains the technical and implementation details for our proposed method.

1.2 LITERATURE REVIEW AND RELEVANT THEORIES

1.2.1 Market Structure Analysis

According to Elrod et al.'s (2002) definition, the goal of market structure analysis is to explain the nature and extent of competition among companies and their products. Using multi-attribute utility theory, researchers can derive market structures by analyzing customers' values for a predetermined set of product attributes (i.e., external analysis) or *ex post* interpretations of derived dimensions from preference or choice data (i.e., internal analysis) (Lee and Bradlow 2011). External analyses presume that researchers know which attributes drive choices; internal analyses instead help identify important dimensions and whether preference data fit the positions of existing brands (Elrod et al. 2002). Elrod et al. (2002) and Myers and Tauber (2011) provide broad overviews of such market structure analysis models.

Classical market structure analysis methods are not without limitations. For example, the process of extracting product attributes through a consumer survey typically is guided by marketing managers, who might focus on different issues than those raised by consumers. Our text analytics approach seeks to take advantage of UGC to compensate for such limitations. By mining product reviews to discover "unseen" product attributes, manufacturers can identify and resolve previously unanticipated problems and add new product attributes that target consumer demand. In addition, in social media environments, the voice of consumers can be collected quickly, frequently, and at a lower cost compared with primary data collection methods. Automated analyses of UGC might increase the return on investment of market structure analysis efforts, such that firms can extend their understanding of customer behaviors.

1.2.2 The Hierarchical Structure of Product Attributes

The hierarchical structure of product attributes has been studied extensively (Johnson 1988; Johnson 1989; Johnson et al. 1992). These studies support the idea that attributes are associated with products, ranging from concrete to abstract. Because of their information processing limitations, consumers prefer to use fewer, abstract attributes that aggregate concrete attributes. To infer attribute relationships, researchers traditionally gather data from human judges, then apply hierarchical clustering to aggregate the attributes (Johnson 1988).

Another theoretical foundation for our proposed attribute hierarchy comes from means-end chain theory (Gutman 1982), according to which consumers associate concrete product attributes with their product usage or consumption situations, which generate benefits for them. The bottom level of the attribute hierarchy contains relatively concrete attributes. Higher levels aggregate these attributes, reflecting how consumers conceptually group attributes according to their usage situations to achieve desired ends or valued states. Hofstede et al. (1999) propose an

international market segmentation model based on this theory, in which consumer choice alternatives get evaluated on the basis of benefits and values instead of physical attributes. The implication for our research is that we can discover consumer usage situations and build an attribute hierarchy using discovered usage situations automatically.

A similar view emerges from market segmentation literature. When segmenting consumer markets, Haley (1968) advises managers to concentrate more on abstract product benefits than on concrete product factors. Although information may be lost in the abstraction process, roughly the same amount of information is contained in a few abstract attributes (Johnson 1984). Abstract attributes are general and especially helpful when conducting analyses of products with heterogeneous concrete attributes. For example, tablet computers have various storage options, and it is intuitive and natural to compare how two tablets can satisfy consumer's storage needs, rather than measuring the usefulness of individual physical attributes such as SD cards and the hard drive.

A few marketing studies discuss ways to extract product attributes automatically from UGC (Archak et al. 2011; Decker and Trusov 2010), though only Lee and Bradlow (2011) examine how to aggregate product attributes in a hierarchical way. They use the term *meta-attribute*, originally proposed by Ghose and Rao (2007), to denote the bundle of product attributes at a higher level. To be consistent, we also use "meta-attributes" to denote higher level, more abstract attributes. However, Lee and Bradlow address only a single layer of meta-attributes; for example, for a digital camera, they identify three product meta-attributes: autofocus, manual focus, and PC connect. These three attributes are separate and controlled by different physical components in a camera, yet autofocus and manual focus likely lie in much closer proximity in consumers' perceptual spaces than do autofocus and PC connect, because

they refer to the same usage situation (i.e., focusing before taking pictures). Our method instead constructs an attribute hierarchy to capture how attributes might be aggregated to multiple higher levels, according to their usage situations, as in Figure 3. We believe this multilevel attribute hierarchy better represents customer usage situations and attribute relationships. The VOC, such as usage situations and attribute synonyms, captured by the attribute hierarchy in turn provides firms with valuable information they can use to design and promote their products.

Insert Figure 3 about here

1.2.3 Text Mining and Sentiment Analysis

Text mining reveals quality information from unstructured text, which is useful, meaningful, and nontrivial (Dörre et al. 1999; Feldman and Sanger 2006). It draws on several subjects, including computational linguistics, information retrieval, machine learning, natural language processing, and statistics. Its use originated in computer science, but today it applies broadly to serve a wide variety of business needs (Feldman et al. 1998; Hu and Liu 2004). Sentiment analysis, also known as opinion mining, infers sentiment polarities by analyzing unstructured text (Liu 2012; Pang and Lee 2008). Recently, both text mining and sentiment analysis have been applied in marketing research. Archak et al. (2011) study the relationship between product attributes and sales of electronic products, and Ghose et al. (2012) combine text mining with crowd-sourcing methods to estimate demand for hotels. Tirunillai and Tellis (2012) demonstrate that product chatter, defined by the magnitude, sentiment, and star ratings of product reviews, can predict firms' stock performance. Decker and Trusov (2010) estimate consumer preferences for product attributes by text mining product reviews, and Onishi and Manchanda (2012) study the predictive power of online blogging in the presence of traditional media.

Although text mining UGC for market structure analyses is a compelling marketing research direction (Fader and Winer 2012), to the best of our knowledge, only two articles address it, and no research combines text mining with sentiment analysis for market structure analysis. Lee and Bradlow (2011) parse product attributes from user-generated pro–con lists and use correspondence analysis (CA) to depict brand distances, according to differences in their attribute counts. Netzer et al. (2012) study how UGC can provide information about competitive market structures by analyzing brand co-occurrences in online forum discussions. Both approaches create market structure perceptual maps with better external validity than conventional approaches and provide information that conventional approaches cannot offer. However, to go beyond the question of whether extant pairs of brands are substitutes or complements, both approaches need additional ad hoc analyses. Netzer et al. (2012) use regression to explain car models, using car characteristics and common discussion terms as explanatory variables. Lee and Bradlow's (2011) CA map also has limited explanatory power, because the axis only reflects relative differences between brands' underlying attribute counts. That is, the market structure provided by CA addresses the importance of attributes but not how they are perceived differently by consumers who evaluate the products.

1.2.4 Contributions

In light of the literature reviewed, our research makes at least three unique contributions. First, our method combines marketing theories, computational linguistics, and text mining to construct the attribute hierarchy automatically. The proposed multilevel attribute hierarchy can capture consumer usage situations and potentially applies to varied marketing research.

Second, we combine the discovery of the market structure and the explanation of the competitive landscape in one model. Instead of using the brand co-occurrence and attribute

frequency data, our method analyzes reviews to reveal consumer sentiments toward attributes and relies on discovered sentiments to draw market insights. We investigate product positions and market segmentation by advancing beyond counts of how often consumers mention brands, products, and their attributes to assess consumers' product usage situations and sentiments.

Third, the proposed method is tailored to deal with free-form reviews. Advances in computational linguistic techniques enable us to analyze the grammatical relationships of review sentences and thereby reveal sophisticated consumer preferences and insights. We validate the proposed method by comparing the empirical results to market structures derived from both extant text mining methods and external data that do not use UGC. Our method grants marketing researchers and practitioners a tool to gain deeper consumer and market insights.

1.3 MARKET STRUCTURE ANALYSIS METHOD

Our proposed method consists of five steps (Figure 4): (1) online review collection, (2) text preprocessing, (3) product attribute extraction, (4) attribute hierarchy identification, and (5) sentiment analysis and perceptual mapping. We provide a brief discussion of these five steps and present the technical and implementation details of our method in Web Appendix A.

Insert Figure 4 about here

1.3.1 Product Attribute Extraction from Reviews (Steps 1–3)

We begin by collecting, cleaning, and organizing product reviews automatically for a product category of interest, which includes removing HTML tags, correcting commonly misspelled words, and extracting relevant details such as posted date and brand. In step 2, we break the unstructured product reviews into linguistic components for text analysis with three tasks: tokenization, part-of-speech (POS) tagging, and dependency parsing. The second step also

identifies the parts of speech and grammatical relationships between words for subsequent analyses.

In the attribute extraction step, natural language processing (NLP) techniques enable us to infer the set of the most salient product attributes from review sentences. Our algorithm improves on Hu and Liu's (2004) paradigm for mining consumer opinions by applying a set of filters on the most common nouns and noun phrases. This step offers a counterpart to traditional attribute elicitation procedures that use individual or group interviews (Steenkamp and Van Trijp 1997). We could use a list of predetermined attributes of interest in the product category, but our method instead uses NLP techniques to identify product attributes automatically, with several valuable implications. First, as Myers and Alpert (1968) note, only a limited set of attributes are really critical to consumers, and text analytics can identify these attributes, as well as their relative importance. Second, according to Lee and Bradlow (2011), consumers may refer to the same attribute using different terms; our method discovers attribute synonyms. Third, the attributes that consumers discuss might be ignored by a traditional approach but can be extracted by mining product reviews.

1.3.2 Attribute Hierarchy Identification (Step 4)

Step 4 constructs the attribute hierarchy. Consumers' perceptions of attribute relationships are measured by their relatedness in product reviews, or *semantic similarities*. According to their semantic similarities, we implement hierarchical clustering to aggregate attributes into a multilevel hierarchy. Then we can choose any level in the attribute hierarchy to obtain a list of meta-attributes, each of which bundles lower-level attributes. The construction of the attribute hierarchy is a form of ontology learning for a product category (Maedche and Staab 2004).

The distributional hypothesis in linguistics provides a theoretical foundation for constructing the attribute hierarchy. Harris (1968, p. 12) suggests that words with similar meanings tend to occur with similar neighbors, such that "The meaning of entities, and the meaning of grammatical relations among them, is related to the restriction of combinations of these entities relative to other entities." Therefore, when consumers express their product experiences using different attributes in reviews, we can infer their usage situations, as well as how the attributes relate to the consumers' needs, motivations, and goals (Netzer et al. 2008).

Consider two tablet reviews: "With the Bluetooth feature you can stream music to speakers," and "My Bluetooth keyboard won't work right now." We can infer that Bluetooth is an attribute that connects with other hardware peripherals, such as speakers and keyboards. The reviews, "The USB supports keyboards, flash drives/external hard drives formatted under FAT," and "The full size USB port lets you plug in an ordinary wired or wireless mouse or keyboard," imply that USB, though usually considered a different product attribute, largely shares its usage situations with Bluetooth. Both are used to connect to peripheral hardware, namely, keyboards. Therefore, we can consider USB and Bluetooth similar and aggregate them into a meta-attribute, pertaining to hardware connection usage situations.

To calculate the semantic similarity of the product attributes identified, we first infer their usage situations from the grammatical relationships of review sentences. In the sentence, "The USB supports keyboards," the subject USB performs the action of support on the direct object keyboards. For each attribute, we quantitatively summarize all grammatical relationships expressed in the reviews with a semantic vector. Each semantic vector contains important grammatical dependency relationships between an attribute and its related action words (i.e., frequently associated verbs). Finally, the similarity between a pair of attributes can be calculated

using a similarity measure (in our method, the cosine distance) of the corresponding semantic vectors. Semantic similarity based on a large collection of text is also called distributional similarity in statistical natural language processing (Lin 1998).

Prior studies in marketing support our approach. The similarity of product attributes relates to their usage situations (Myers and Shocker 1981). Usage situations also play a critical role in consumers' perceptions of products (Koukova et al. 2012) and determine the benefits that the consumer seeks (Srivastava et al 1984). In addition, from a consumer perspective, the benefits provided in different usage situations are what consumers seek from products and what define product markets (Day et al. 1979; Srivastava et al. 1984). Put another way, when two product attributes relate to the same usage situation, they are perceived as providing similar benefits (Ratneshwar and Shocker 1991).

We adopt hierarchical clustering to construct the attribute hierarchy, as shown in Figure 3, Panel b, for several reasons. First, cluster analysis has served as a fundamental tool in marketing research. Srivastava et al. (1981) suggest using hierarchical clustering with product usage data to explain variance in product categorizations. Lee and Bradlow (2011) use cluster analysis to group product attributes in pro–con lists from consumer-generated reviews, though the similarity measure we use is based on a different theory. Second, in knowledge discovery literature, cluster analysis supports ontology learning from text (Buitelaar et al. 2005). In this research, the attribute hierarchy represents the domain ontology, with concepts (attributes) and their relationships in a product category. Third, using a dendrogram to represent the attribute hierarchy, we can obtain meta-attributes with different levels of abstraction by cutting the dendrogram at a specific level, according our analysis needs.

1.3.3 Sentiment Analysis and Perceptual Mapping (Step 5)

In the last step, we summarize consumers' opinions (i.e., sentiment polarities) toward product attributes using sentiment analysis techniques. Unlike previous research that used brand co-occurrence data (Netzer et al. 2012) or attribute frequency data (Lee and Bradlow 2011), we derive consumers' sentiments, which can directly explain why they perceive or evaluate competitive products differently. The attribute hierarchy and meta-attributes also provide an attribute simplification framework that summarizes relatively sparse consumer sentiments toward product attributes.

Sentiment classifiers are machine learning algorithms that can automatically detect consumers' sentiments toward product attributes as positive or negative. Consumer' sentiments are measured as follows: Let MA_{ij} be the meta-attribute j for a brand i, which is a bundle of lower-level attributes. We find the subset of all review sentences that mention a lowest-level product attribute in MA_{ij}. Then we feed these sentences into a sentiment classifier that we have trained by tagging a subset of sentences. The classifier outputs P_{ij}, which is the number of review sentences with positive sentiments, and N_{ij}, which is the number of review sentences with negative sentiments. We choose the ratio P_{ij}/N_{ij} as the sentiment score[1] of MA_{ij}. Web Appendix A4 elaborates on the implementation of sentiment analysis and training of sentiment classifiers.

With these derived consumer sentiments, we apply multidimensional scaling (Green and Carmone 1969) to generate the market structure perceptual map. Multidimensional scaling (MDS) is a popular multivariate technique to explore relationships among brands in terms of

[1] We found no other marketing studies that compare sentiment measures using sentence polarity counts. We investigated other sentiment rating ratios such as $P_{ij}/(N_{ij}+P_{ij})$ and $\frac{P_{ij}}{T_{ij}}$, where T_{ij} is the total number of sentences that contain meta-attribute j in a brand i. The MDS analysis was robust to the rating ratio chosen.

consumers' perceptions and preferences (Myers and Tauber 2011); it transforms consumer judgments of similarity or preferences for brands into distances represented in multidimensional space. The MDS maps show the relative positioning of all brands (Carroll and Green 1997). We advance Netzer et al.'s (2012) work by expanding MDS beyond its usual reliance on data from surveys and experiments to include UGC. We show empirically that sentiment analysis on free-form reviews, coupled with a traditional MDS technique (i.e., sentiment MDS approach), has great potential for constructing perceptual maps with high face and external validity.

1.4 EMPIRICAL ANALYSIS AND EVALUATION

1.4.1 Empirical Study of Tablet Computers

With a prototype system developed according to our method, we conducted an empirical study to analyze tablet computer reviews collected from Amazon.com. Tablet computers are a representative product of digital convergence (Yoffie 1996) and serve many functions in consumers' lives. Because of the various components involved, it can be difficult for manufacturers to determine the perfect mix of attributes to include in a tablet computer, and many traditional PC manufacturers that have tried to enter this market have suffered disappointing results. For example, Hewlett-Packard halted production of its Touchpad less than two months after its launch and sold its remaining inventory at deep discounts, after recognizing that the product was out of touch with consumers (Sloane 2011; Tsuruoka 2011). To offer manufacturers new insights, we consider how online product reviews might shed light on this quickly evolving market. Amazon's product reviews reasonably represent UGC available online, because Amazon is among the largest and most successful e-commerce websites. We

implemented web-scraping software to collect more than 20,000 tablet reviews and associated product information in July 2012.[2]

After text preprocessing to clean the collected reviews, we used Stanford CoreNLP to complete the text preprocessing tasks, including POS tagging and dependency parsing. The data set includes 190 brands and 703 tablet products, covered in 20,157 reviews. Each review contained an average of 13 sentences, and each review sentence consisted of 17 words on average. In total, we analyzed 270,497 review sentences and 4,578,180 words.

1.4.2 Attributes and Attribute Hierarchy of Tablet Computers

We provide nine noun phrases and their related linguistic measures in Table 1 to illustrate the method for extracting product attributes. The support measure indicates the frequency of the noun phrases. Pure support of a noun phrase reflects the proportion of reviews in which the phrase appears but not any superset of the phrase. A low pure support-to-support ratio suggests that the noun phrase by itself has little meaning. The likelihood ratio indicates the relative importance of a noun phrase with respect to the product context. For definitions of these measures, see Web Appendix A2.

We set the pure support-to-support ratio threshold to .1, and the likelihood ratio cutoff to 2,000. Because of their low likelihood ratios, we excluded the terms *one* and *time*. These noun phrases are not specific enough to tablet computers. The noun phrases *market* and *life* were eliminated due to their small pure support-to-support ratios, indicating that they were not prominent by themselves in the reviews. Instead, *android market* and *battery life* were identified

[2] We excluded Kindle Fire reviews, because Amazon is the dominant channel of distribution for these tablets and hosts a disproportionate number of product reviews, which might distort the product attribute identification and provide biased sentiment.

as tablet attributes. We manually filtered out the noun phrase *tablet*, because it refers to the product category.

Insert Table 1 about here

At a support threshold of .00385 (i.e., .385% of the reviews mentioned these attributes), 93 attributes could be extracted. This threshold may be adjusted subjectively, depending on the size of the data set and how exhaustive the researcher wants the market structure analysis to be. We experimented with different thresholds and concluded that those in Table 2 worked best for our data. We elaborate on the attribute extraction in Web Appendix A2.

Insert Table 2 about here

With these 93 tablet attributes, we calculated semantic similarity between each pair by first determining the semantic vectors for every product attribute, then the cosine distance for each pair. Each semantic vector contains the pointwise mutual information (PMI) between an attribute and its dependence words. We used add-one smoothing to calculate the PMI, defined as $\text{PMI}(word_1, word_2) = \log_2 \frac{C(word_1, word_2)+1}{(C(word_1)+1)(C(word_2)+1)}$, where $C(word_1)$ is the frequency of sentences that contain $word_1$.

For illustration, Table 3, Panel a, lists the semantic vectors of three product attributes: *YouTube*, *webcam*, and *USB cable*. We show only the top PMI scores for each product attribute's vector. These entries carry valuable information about product attributes and are representative of the most common usage situations. The entry [*dobj, charge*] for *USB cable* shows that in the product reviews, *USB cable* was the direct object in a dependency relationship with *charge*, and the PMI score of 3.4966 indicated its high information content. Similarly, other usage situations involving *USB cable* included connecting and recognizing other devices and whether it was included with the product, would break easily, could be inserted, and was recognizable. By

translating the usage situations of product attributes into numerical semantic vectors, we quantitatively assessed similarities across them, as we show in Table 3, Panel b. We computed a 93×93 similarity matrix using the cosine similarity measure; the results provided the input for the cluster analysis to construct the attribute hierarchy.

Insert Table 3 about here

We present the dendrogram of the tablet's attribute hierarchy in Figure 5, obtained by implementing an agglomerative procedure on the similarity matrix. We can choose any particular number of meta-attributes by cutting the dendrogram at an appropriate level. For our study, we chose seven meta-attributes (i.e., seven clusters), for two reasons. First, the quantitative evidence gathered from several unsupervised cluster evaluation measures indicates that the seven-cluster solution is "natural" and fits the data well (for details, see Web Appendix A3). Second, these seven meta-attributes effectively summarize how consumers evaluate tablet computers, as we demonstrate subsequently by comparing them with expert guides and participant evaluations.

Insert Figure 5 about here

Table 4 presents the seven meta-attributes: multimedia, storage, operating system, connectivity, hardware specification, everyday activity, and user interface. We labeled them according to their aggregated attributes and common usage situations (i.e., associated verbs). The labeling requires human assessment, which is a limitation shared by all learning hierarchies derived from textual data (Cimiano and Staab 2005). We compiled common usage situations for each meta-attribute from their individual attributes' dependency relationships with the highest PMI scores.

Insert Table 4 about here

We could cut the dendrogram of product attributes at a higher level, which would result in fewer meta-attributes, or at a lower level, which would provide more meta-attributes. Consider the user interface meta-attribute in Figure 6. A meaningful hierarchical structure exists within it, such that we can differentiate the group of attributes at the top, corresponding to the keyboard interface, from the larger group of attributes at the bottom, which correspond to the screen interface. At the highest level, we observe that all attributes pertaining to multimedia are separated out as one of the two major meta-attributes. This significance of multimedia in the attribute hierarchy is not surprising; many studies highlight media consumption as a primary function of tablet computers (ABI Research 2012), and tablets are starting to replace traditional computers and televisions as dominant digital media consumption devices (Steel 2013; Walsh 2011).

Insert Figure 6 about here

Our method also recognizes attribute synonyms that consumers use to refer to the same product attribute. For example, *wifi* and *wi-fi*; *microsd*, *microsd slot*, and *sd slot*; and *web browsing* and *web surfing* grouped in the same clusters at the first level of aggregation. Consumers used these noun phrases interchangeably in their tablet reviews. Attribute synonyms and usage situations help facilitate communication with target consumers, by using their own language.

1.4.3 Validation of the Attribute Hierarchy

To evaluate the tablet's attribute hierarchy, we first compared the higher-level meta-attributes with several expert buying guides, which usually mention the aspects that are most pertinent to buying decisions. Similar to Lee and Bradlow (2011), we verified whether consumer-generated reviews revealed product attributes not found in expert guides, and vice

versa, as we detail in Table 5, Panel a. We computed precision P (Salton and McGill 1983) as the number of automatically generated attributes also used by experts in their guides and recall R as the count of the number of attributes and levels named in these guides that were automatically extracted by our empirical study. Assume X is the set of attributes in the reviews and Y is the set of attributes in buying guides. Then P and R are defined as follows: $P = |X \cap Y| / |X|$, and $R = |X \cap Y| / |Y|$. In Table 5, Panel b, the first row indicates the precision, and the second indicates the recall; analyzing reviews yields higher recall than attributes mentioned by expert guides. That is, our method reveals nearly all the attributes that expert guides do. In addition, consumer-generated reviews include product attributes omitted from individual expert guides.

Insert Table 5 about here

We also assessed the quality of the attribute hierarchy with a web-based survey (see Cimiano and Staab 2005). We asked 179 students in a Midwestern U.S. university to evaluate the relationships among the seven meta-attributes and individual attributes (see Web Appendix B). In the survey, students considered random pairs of meta-attributes and attributes and rated the level of correspondence on a five-point scale, where 5 represents the highest level of correspondence. In order to evaluate both discriminant and convergent validity of the meta-attributes, a product attribute may or may not correspond to a particular meta-attribute in the survey.

In Table 6, we summarize the percentage of ratings greater than or equal to 3 for each meta-attribute, which provided the similarity measure between lower-level attributes and a higher-level meta-attribute. We include both the similarity scores for attributes within the meta-attribute clusters generated by our method and scores for attributes beyond the meta-attribute clusters. The relatively high percentages show that the correspondence between meta-attributes

and attributes is reasonable, according to the survey participants. In addition, the lower similarity measures suggest that our method effectively differentiates attributes unrelated to the abstraction represented by the meta-attributes.

Insert Table 6 about here

1.4.4 Sentiment Analysis

To conduct the sentiment analysis, we randomly selected and manually tagged 2,000 review sentences in three sentiment groups: positive, negative, and neutral. The percentage of agreement between the two human raters was 84%, and the interrater reliability measure Cohen's kappa (Cohen 1968) reached .81. The two raters were fairly consistent in detecting sentiment polarities with good interrater reliability, but we cannot expect higher accuracy by any automatic sentiment analysis method, because of the inherent ambiguity of the language in free-form reviews. With the tagged reviews as a training set, we evaluated four common machine learning methods in terms of their sentiment classification: maximum entropy, classification tree, naïve Bayes, and support vector machine (SVM). Each method trained two sentiment classifiers to detect positive and negative sentiments automatically. We used bagging (Breiman 1996) to enhance the accuracy of these classifiers, training individual classifiers on bootstrapped samples, and we used majority voting from multiple classifiers to determine the final prediction. The SVM provided the best overall accuracy, as we discuss in detail in Web Appendix A4.

1.4.5 Perceptual Mapping of Tablet Brands

To obtain the market structure perceptual map for tablets, we chose 15 brands with the most product reviews and conducted sentiment analysis to obtain the sentiment scores of the seven meta-attributes. We used the root mean square normalized ratio of $\frac{\text{number of positive sentences}}{\text{number of negative sentences}}$ to represent sentiment toward the 15 brands' seven meta-attributes. We

also added 1 to both the numerator and denominator. The sentiment scores for 6 major tablet brands appear in Figure 7. We next generated a MDS map using the sentiment matrix. According to Figure 8, reducing the space to two dimensions is sufficient (R^2 = .804). Figure 9 contains the MDS map of the 15 brands. Finally, we used MDS coordinates to run a *k*-means clustering and determine the potential market segmentation. We chose a range of clusters to perform the *k*-means analysis; four clusters provided the best visual presentation. The brands in the same cluster appear in a dotted circle; Asus was the only brand in its cluster.

Insert Figures 7, 8 and 9 about here

The locations of major brands in Figure 9 exhibited face validity. First, most lower-end brands clustered together in the right side of the MDS map. These brands represented companies whose main products were not computers, as well as Dell and Lenovo, whose main targets were desktop and laptop users and which entered the tablet market late. Second, in two other clusters (assuming the center of the map is an average brand; Torres and Bijmolt 2009), we found well-distinguished, leading brands, such as Apple and Asus. Third, Apple and Hewlett-Packard were the only manufacturers to use their own operating systems, and both were distant from other brands. These results are consistent with the IDC (2012) tablet forecast report, which predicted a dominant share for independent operating systems (53.8%) over Android (42.7%).

We regressed each brand's X1 and X2 coordinates in the MDS map on its meta-attributes' sentiment scores. The greatest estimates on the X1 axis corresponded to Multimedia, Operating System, and User Interface, whereas the X2 axis combined Connectivity, Hardware Specification, and Everyday Activity. Therefore, we interpreted the X2 axis as tablets' hardware components, and the X1 axis represented software and interactive components. Most of the 15 brands did not differ much on the X2 axis (i.e., hardware components), suggesting that tablet

manufacturers could focus on enhancing the consumer's experience with a given hardware design to distinguish themselves from competitors. The tablet market leader Apple stood out on both dimensions. Another prominent case was Asus, with its Transformer line. Asus is known for pushing boundaries and including the best hardware, such as by introducing the world's first tablet with a quad-core processor (Asus 2011). From the MDS map and its sentiment scores, it appears that Asus's commitment to the best hardware has gained consumers' appreciation for the overall use experience too. Tablet manufacturers should align their efforts on both dimensions, rather than considering them independent directions, to improve their product development.

To derive additional insights about brand proximity, we collected information about eight product attributes; the median attribute values of all options appear in Table 7. We used *k*-means cluster analysis to segment the 15 brands, using numerical attribute levels in product attribute space. Three distinct clusters emerged: Acer, Asus, HP, Motorola, Samsung, and Toshiba belong to the first cluster; Apple, Archos, Dell, Le Pan, and Lenovo belong to the second; and Coby, Pandigital, Velocity, and ViewSonic constitute the third cluster. Comparing the clusters generated from these nine attributes with the MDS market structure perceptual map constructed by the sentiment matrix yielded some interesting differences. For example, the four brands in the third cluster clearly target the budget market, with products on the lower end of the hardware spectrum. In the MDS map, the traditional PC powerhouses Lenovo and Dell are closely associated with these lower-end brands in brand proximity. Referring to the consumer sentiments in Figure 7, Lenovo and Dell have noticeably unfavorable scores for multimedia, user interface, and operating systems. This suggests that though their products are equipped with mainstream hardware, Lenovo and Dell's old-style, Windows-based, convertible laptops with lackluster attempts at building Android tablets were not well perceived by consumers in mid-2012. Such

results offer an explanation for why Dell and Lenovo launched new product lines (Newman 2013) and adopted the Windows 8 operating system, with a new touch user interface and improvements in various areas, including multimedia (Sinofsky 2012). Our empirical study shows that UGC alone may not provide an exact prescription for manufacturers, but when combined with other market intelligence resources; UGC can be effective for supporting managerial decisions.

Insert Table 7 about here

1.4.6 Validation of Tablet Computers' Market Structure

To quantitatively validate the derived tablet market structure, we compare the perceptual map with external data sources. Tablets constitute a relatively new product category, and we are not aware of any publicly available brand-switching or industry-scale survey data, which Netzer et al. (2012) used to assess external validity. Therefore we resort to two external proxy measures and compare our results with the market structure derived from them.

The first data set is the Factiva news database, which contains news published by top media outlets. We used the keyword "tablet" and manufacturers' stock tickers to formulate our search queries. We searched publications on Factiva from April 1, 2010, to July 31, 2012, which represents the time frame of our review data set. This approach has been used to establish simultaneous cooperation and competition between firms (Gnyawali and Park 2011); in the tablet market for example, many manufacturers have the same architectures and operating systems. As a synopsis, we used an information similarity measure, normalized Google distance (NGD). The NGD for two brands, according to the formula derived by Cilibrasi and Vitanyi (2007), is:

$$NGD(x,y) = \frac{\max[\log f(x),\log f(y)] - \log f(x,y)}{\log M - \min[\log f(x),\log f(y)]},$$

where $f(x)$ is the number of news releases with "tablet" as a keyword and the manufacturer ticker x, and M is the total number of news releases with "tablet" as a keyword. The greater the NGD, the less related two brands are; it offers a good measure of semantic relatedness based on web search results (Veksler et al. 2008). To increase the statistical power of the comparison, we obtained an 11×11 NGD matrix by excluding some smaller manufacturers that did not return any results. The correlation between the normalized NGD matrix derived from the Factiva news database and the normalized consumer sentiment distance matrix was .633 ($p < .001$), which is relatively high and significant. All p-values were estimated using Mantel's (QAP) test.

In addition, we compared our results with a data set similar to the market structure surveillance method proposed and validated by Netzer et al. (2012). We calculated the brand co-occurrence similarity measure, using forum discussion data, which provided a proxy for brand switching and consideration set data. Netzer et al. (2012) have shown empirically that brand co-mentions correlate closely with both brand switching and consideration set data. We chose the "What Tablet PC Should I Buy?" forum of tabletpcreview.com, which contains more than 5,171 threads and 33,856 messages by customers discussing choices, features, and options among tablet PC selections. This data set covers brands that consumers consider seriously when making purchase decisions, which reflects the traditionally used consideration set (Hauser and Wernerfelt 1990) and therefore provides an even more specific measure of consumers' perceptual space than the more general forum discussions analyzed by Netzer et al. (2012).

We searched for brand name occurrences and co-occurrences of tablet brands using a function provided by the forum and retained the results for the time frame of our review data set. The lift of two terms x, y would be defined as $\frac{P(x,y)}{P(x)P(y)}$. Again, the correlation between the consumer discussion brand co-occurrence matrix and the distance matrix from consumer

sentiments was very high ($\rho = .714, p < .001$), in further support of our method's comparative validity. We also compared the correlation between the normalized lift matrix generated from the Factiva news search and the forum discussion. The correlation of .702 suggested that the two external proxy measures we used had high internal consistency.

Finally, we transformed the 20,157 tablet computer reviews to create a structure similar to that used by Lee and Bradlow (2011); our sentiment MDS approach produced a more accurate market structure. Lee and Bradlow (2011) conducted correspondence analysis (CA) with two-way product attribute frequency counts, then compared the CA map conceptually with product market share and brand strategies, with no reported quantitative measure. To transform our tablet reviews into a two-way attribute frequency count, we tallied the seven meta-attribute frequencies from the reviews for the major brands, without conducting further sentiment analysis. We used CA to depict the brand by attribute count matrix (see Figure 10). To compare the Euclidean distance matrix between pairs of brands on the CA map, we used the distance matrix generated using external market structure measures. The correlation between the CA distances with Factiva news relatedness was .450, and the correlation between CA and the forum discussion was .437, both statistically significant at $p = .05$. These results show that Lee and Bradlow's (2011) method can produce a somewhat representative map. However, because these correlations are much lower than that of our sentiment MDS approach, the CA approach is less accurate in revealing market structures than our approach for market structure analysis.

Insert Figure 10 about here

1.4.7 Detecting Market Structure Shifts

Both consumers and manufacturers in the tablet market face constant flux and evolving technology. Monitoring consumer sentiments provides manufacturers a tool to reassess their

market positions and competitive dynamics in light of new products and changing consumer preferences. For a comparison with the previously generated market structure of for tablets, we collected an additional set of 25,738 reviews of 383 products launched between July 2012 and September 2013. Figure 11 depicts the MDS map generated from this additional data set. As it reveals, the relative positions of major brands were similar compared with the original map, with a few notable exceptions. For example, by launching new Windows 8 tablets such as ThinkPad Helix, Yoga, and Twist, Lenovo started to distinguish itself from the budget brands. We statistically tested for the cause of this market structure shift using a bootstrapped procedure and resampled the review sentences 1000 times. We compared Lenovo with three budget brands (Coby, Archos, and Velocity) on the seven meta-attributes, according to the bootstrap distributions of their sentiment differences. Table 8 shows the comparison before and after July 2012; in the pre-July 2012 period, no significant difference (p = .05) emerged for five of seven meta-attribute sentiments between Lenovo and the three budget brands. However, after July 2012, six of the seven meta-attributes for Lenovo exhibited statistically significantly higher (p = .05) sentiment measures. Thus, switching its tablet product lines toward Windows systems helped Lenovo regain consumer satisfaction. This consumer-centric evaluation of the effect of new product strategies grants manufacturers access to feedback directly from consumer-generated product reviews.

Insert Figure 11 and Table 8 about here.

1.4.8 Robustness Checks

Other Clustering Methods and Number of Clusters. We evaluated the robustness of the derived meta-attributes by comparing our chosen hierarchical clustering method with another

popular clustering method, *k*-medoids.[3] The *k*-medoids method requires the number of clusters k to be known a priori, so we compared the seven meta-attribute choice given by these two methods using the ontology similarity measures proposed by Maedche and Staab (2002). In particular, if two attribute hierarchies (ontologies), O_1 and O_2, have been discovered using two methods, with c_i representing individual product attributes in a collection of N attributes denoted as C, then the taxonomic overlap ($\overline{TO}$) can be computed as

$$\overline{TO}(O_1, O_2) = \frac{1}{N} \sum_{c_i \in C} TO(c_i, O_1, O_2),$$

where $TO(c_i, O_1, O_2) = \frac{|\text{Intersection of attributes sharing the same cluster with } c_i \text{ in } O_1 \text{ and } O_2|}{|\text{Union of of attributes sharing the same cluster with } c_i \text{ in } O_1 \text{ and } O_2|}$. If $\overline{TO}$ equals 1, the two attribute hierarchies are exactly the same; any value greater than .5 can be interpreted as high agreement. In our study, we found $\overline{TO}$ with *k*-medoids equal to .57. Table 9 presents the $\overline{TO}$ measures for the seven meta-attributes; other than Everyday Activity, a generic bundle of lower-level attributes, the meta-attributes were very robust to clustering methods.

Inset Table 9 about here.

We also examined the robustness of the seven-cluster solution by considering how the number of meta-attributes affected the market structure analysis results. We conducted a sentiment analysis by cutting the dendrogram in Figure 5 at the four- and ten-cluster levels, then comparing the brand distance matrices with the distance matrix generated at the seven-cluster level. Mantel's tests showed correlations of .853 and .810, respectively, both significant at p = .01. That is, the results remained similar with more or fewer meta-attributes, which implies researchers have the flexibility to decide how fine-grained to make their market structure analysis at the attribute level.

[3] Although *k*-means is a better known clustering method, it does not work with non-Euclidean distance measures.

Effect of Machine Learning Algorithms for Sentiment Classification. Beyond machine learning techniques, marketing researchers often use a lexicon approach for sentiment classification. In the lexicon approach, a precompiled dictionary of positive and negative words indicates the sentiment polarity of a textual document by their difference (e.g., Berger and Milkman 2012). Table 10 compares the accuracy of SVM and the lexicon approach using Hu and Liu's (2004) sentiment lexicon, containing approximately 6800 words; SVM provides better accuracy and F1 scores for detecting both positive and negative sentiments than the lexicon approach. The improvement is especially noticeable in the precision for positive sentiments and recall for negative sentiments. The application of SVM for sentiment classification thus is better at discovering negative opinions and reduces noise in positive opinions.

Insert Table 10 about here.

The advantage of the lexicon approach is that it is easier to code and offers faster computation speed. However, with many types of well-optimized, open source machine learning software available, the barrier has been lowered for marketing researchers and practitioners to use these state-of-the-art techniques for sentiment analysis.

Number of Reviews. As a further robustness check, we looked at how sparser data might affect the sensitivity of market structure maps. We used a robustness check procedure similar to that described by Netzer et al. (2012) by randomly sampling from 10% to 90% of the original data without replacement. We calculated the correlation between the distance matrix derived from brand sentiments using the full data set with the distance matrices generated from the subsets. Figure 12 illustrates the results. All correlation coefficients were significant at $p = .01$. The results suggest that our method can lead to similar market structures using sparser data.

Insert Figure 12 about here.

In addition, we investigated the effect of imbalanced reviews among brands. For each brand, we simulated the size of reviews, independently uniformly distributed from 50% to 300% of the original size, through random sampling with replacement. We repeated the process 50 times and found that the mean correlation between the distance matrices from resampled data to the distance matrix from the original data was .84 (SD = .10), suggesting that our method was robust to considerably imbalanced review distributions.

Alternative Measures of Brand Similarity. Crucial to the validation of the derived MDS map is that we use NGD and lift as information similarity measures for brands' "true" similarity. We therefore compared our chosen measures with other commonly used similarity measures: pointwise mutual information (PMI) and Salton cosine. We defined PMI previously, and the Salton cosine is $Cosine_{ij} = \frac{X_{ij}}{\sqrt{X_i X_j}}$. Table 11 presents the correlation between the market structure maps produced using our method and Lee and Bradlow's (2011) CA with the two external data sources, computed with four alternative similarity measures. The results were consistent with our previous conclusions. Our method produced a more meaningful market structure map than the CA approach using attribute counts, demonstrated by higher correlations with the external sources, regardless of the similarity measures used.

Insert Table 11 about here.

1.5 DISCUSSION AND FURTHER RESEARCH

1.5.1 Discussion

Every firm strives to develop new products that can differentiate it from competitors in ways recognizable to their target consumers. However, for a fast evolving and dynamic product market, traditional methods may be costly and lag in identifying consumer preferences and

sentiments, causing manufacturers and brands to struggle. What can product development managers do to stay ahead of the game when it comes to product competition and innovation? Consumer-generated product reviews provide a valuable information source. We develop and evaluate an innovative market structure analysis method to address challenges with analyzing consumer-generated product reviews.

Table 12 compares our method with extant market structure analysis methods. The advantages of analyzing product reviews for market structure analysis already have been established by Lee and Bradlow (2011) and Netzer et al. (2012). With our method, we seek to compensate for and expand their text analytic approaches to free-form reviews.

Insert Table 12 about here.

In addition to being able to provide a market structure with high validity, our method offers several advantages. It can extract salient product attributes from free-form reviews, and by analyzing attributes' usage situations, it constructs a multilevel attribute hierarchy automatically. By referring to this attribute hierarchy as an attribute simplification for sentiment analysis, our method offers a principled and effective approach to building market structures according to consumer sentiments toward product attributes. Neither the attribute-count method proposed by Lee and Bradlow (2011) nor the co-mention of brands method proposed by Netzer et al. (2012) can provide direct answers to questions such as "Why are the two products different on the perceptual map?" or "What should brand A do better to compete with brand B?" Our method generates results that align better with existing collections of market structure models. That is, the market structure derived by our method does more than simply reflect the extent to which pairs of brands are similar; it provides deeper insights regarding the value customers attach to attributes (Elrod et al. 2002) through attribute-based sentiment scores.

1.5.2 Marketing Implications

Our approach and method provide marketing practitioners with several new perspectives with regard to employing user-generated content.

Mining the Voice of Consumers for Effective Advertising

Firms' market communication might be informative only from their own perspective, leaving consumers poorly informed about the quality and attributes of newer products. Yet advertising can emphasize, and human information processing capability can recognize, only a limited number of attributes, rather than every benefit and value associated with a product (Mayzlin and Shin 2011). ConsumerReport.org lists 19 physical attributes for Samsung Galaxy Tab 10.1, but Samsung could not possibly describe each of these attributes clearly in a 30-second advertisement, so it would need to focus on a few significant attributes to differentiate its product from competitors'. The sentiment analysis associated with our method can reveal the most important attributes in light of market competition. Firms can communicate strategically with their target consumers using the significant attributes, in consumers' own language. Then they can adjust their advertising strategy throughout the product life cycle by frequently mining product reviews and detecting shifts in the market structure.

New Product Development with Conjoint Analysis

Manufacturers often use conjoint analysis to evaluate new products prior to introduction, seeking levels of various attributes that can maximize a given objective function, such as market share. The analysis usually involves two stages: Identify consumers' preference structures, usually by estimating the utility partworth through choice-based conjoint analysis (Green et al. 1981; Green and Rao 1971), then solve the optimization problem by choosing the best combination of attribute levels. In choice-based conjoint analysis (Su 2008), the number of

parameters to be estimated depends on the number of attributes. More data are required if the number of parameters is large, and the data collection task may become unmanageable. Moreover, consumers usually do not evaluate the objective attribute levels directly (Nelson 1999), and the optimization problem is NP-hard (Kohli and Krishnamurti 1989), such that most algorithms are efficient only if there are relatively few attributes. Our method of mapping concrete attributes to meta-attributes allows product development managers to avoid overwhelming numbers of attributes and provide better assessments. The sentiment analysis in turn reveals how customers perceive these attributes and can be used in conjoint profiles to help companies find customers' ideal points. Beyond applications in traditional conjoint models, Netzer et al. (2008) note other promising applications of meta-attributes, such as hierarchical Bayesian models for preference evaluation (Luo et al. 2008) and recommendation systems (De Bruyn et al. 2008).

Brand Relationships and Positioning

Brand managers tend to rely on firm-specific information obtained from media stories and site visits or from competitor reports, which are available with low frequency. In contrast, consumer-generated product reviews can be collected at a relatively high frequency, from multiple sources, with extra dimensions such as time, product series, and reviewer demographic information. Brand managers can gain advantages by examining consumer–brand relationships and foster brand loyalty by addressing issues revealed in consumer sentiments. In addition, firms can ensure customer retention by actively adjusting their brand positioning according to how consumers evaluate the functional benefits rather than specific, concrete attributes.

1.5.3 Limitations

Any method using UGC comes with challenges and limitations. The sparse, noisy, unstructured data, compared with traditional data sources, means that our method works better with a large data set. Other natural language processing methods may alleviate the sparsity issue by mining implicit product attributes that are not described using simple noun phrases. Another limitation is that our method only collects reviews from one data source. Integrating product reviews from different sources and languages is an important topic to investigate.

Various unsupervised machine learning techniques in the attribute extraction and hierarchy identification require some human intervention. We chose the tuning parameters according to our best judgment, which may suffer from human decision biases, such as confirmation bias and individual differences. Validating the results of unsupervised methods such as hierarchical clustering is challenging, because the "gold standard" is hard to define (Sabou 2005).

Sentiment analysis on the sentence level remains an ongoing computer science challenge. The bag-of-words model has suffered binary classification accuracy limits of approximately 80% for several years (Socher et al. 2012), which may be the inherent variance for measuring consumer sentiments. However, a recent study using deep learning (Socher et al. 2013) pushed the limit to 85%, such that it may provide more accurate sentiment measurements.

1.5.4 Further Research Directions

Finally, the findings point to interesting research options. First, according to Elrod et al (2002), marketing structure analysis can be conducted at the individual level, due to consumer heterogeneity. Consumer-generated reviews combined with reviewer demographic information can support the construction of individual-level market structures. For decades, firms sought

competitive advantages almost exclusively in activities related to new product markets. Individual consumer-level market structures could offer road maps toward obtaining more sustainable competitive advantages.

Second, markets are dynamic. Periodic assessments of static market structures can track changes, with the implicit assumption that the market is close to a stable equilibrium; however, many markets are characterized by regular product changes, entries, and exits. Firms are keen to determine their dynamic evolution, creating the need to investigate dynamic market strutures.

Third, research should consider alternative structures among attributes. Our hierarchical clustering approach produces disjoint clusters (Feldman and Dagan 1995), which balances the trade-off between the complexity of human conceptions and the necessary structural simplification. A product attribute might have a wide range of usage benefits; for example, a USB cable can connect peripherals, support charging, or help transfer files. Recent advances in analytics allow researchers to explore other structures of product attributes, beyond hierarchical ones, such as latent Dirichlet allocation (Blei et al. 2003) to model different usage situations or model-based clustering (Fraley and Raftery 2002) to allow overlapping clusters of product attributes.

In summary, our study enhances automatic market structure analysis by proposing and validating a new framework for analyzing product reviews using advanced NLP and machine learning techniques. Extensive study and testing is still needed though, considering the volume and richness of UGC.

REFERENCES

ABI Research (2012), "Mobile's Role in a Consumer's Media Day: Smartphones and Tablets Enable Seamless Digital Lives."

Archak, N., A. Ghose, and P.G. Ipeirotis (2011), "Deriving the pricing power of product features by mining consumer reviews," *Management Science*, 57 (8), 1485-509.

Asus (2011), "ASUS Announces the Eee Pad Transformer Prime, the World's First Tablet Featuring the NVIDIA Tegra 3 Quad-Core Processor," [available at http://www.asus.com/News/B780DjsZhrYc9Lts/].

Berger, J. and K.L. Milkman (2012), "What makes online content viral?," *Journal of Marketing Research*, 49 (2), 192-205.

Blei, D.M., A.Y. Ng, and M.I. Jordan (2003), "Latent dirichlet allocation," *Journal of Marketing Research*, 3, 993-1022.

Breiman, L. (1996), "Bagging predictors," *Machine learning*, 24 (2), 123-40.

Carroll, J.D. and P.E. Green (1997), "Psychometric methods in marketing research: Part II, multidimensional scaling," *Journal of Marketing research*, 34 (2), 193-204.

Chen, H., R.H.L. Chiang, and V.C. Storey (2012), "Business intelligence and analytics: From big data to big impact," *MIS Quarterly*, 36 (4), 1165-88.

Cilibrasi, R.L. and P.M. Vitanyi (2007), "The google similarity distance," *IEEE Transactions on Knowledge and Data Engineering*, 19 (3), 370-83.

Cimiano, P. and S. Staab (2005), "Learning concept hierarchies from text with a guided hierarchical clustering algorithm," *Proceedings of Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods, ICML05*, 6-16.

Cohen, J. (1968), "Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit," *Psychological bulletin*, 70 (4), 213-20.

De Bruyn, A., J.C. Liechty, E.K. Huizingh, and G.L. Lilien (2008), "Offering online recommendations with minimum customer input through conjoint-based decision aids," *Marketing Science*, 27 (3), 443-60.

Decker, R. and M. Trusov (2010), "Estimating aggregate consumer preferences from online product reviews," *International Journal of Research in Marketing*, 27 (4), 293-307.

Doan, A., R. Ramakrishnan, and A.Y. Halevy (2011), "Crowdsourcing systems on the world-wide web," *Communications of the ACM*, 54 (4), 86-96.

Dörre, J., P. Gerstl, and R. Seiffert (1999), "Text mining: finding nuggets in mountains of textual data," *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 398-401.

Elrod, T. and M.P. Keane (1995), "A factor-analytic probit model for representing the market structure in panel data," *Journal of Marketing Research*, 32 (1), 1-16.

Elrod, T., G.J. Russell, A.D. Shocker, R.L. Andrews, L. Bacon, B.L. Bayus, J.D. Carroll, R.M. Johnson, W.A. Kamakura, and P. Lenk (2002), "Inferring market structure from customer response to competing and complementary products," *Marketing Letters*, 13 (3), 221-32.

Fader, P.S. and R.S. Winer (2012), "Introduction to the special issue on the emergence and impact of user-generated content," *Marketing Science*, 31 (3), 369-71.

Feldman, R. and I. Dagan (1995), "Knowledge Discovery in Textual Databases (KDT)," *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95)*, 95, 112-17.

Feldman, R., M. Fresko, Y. Kinar, Y. Lindell, O. Liphstat, M. Rajman, Y. Schler, and O. Zamir (1998), "Text mining at the term level," *Principles of Data Mining and Knowledge Discovery*, 65-73.

Feldman, R. and J. Sanger (2006), *The text mining handbook: advanced approaches in analyzing unstructured data*. New York: Cambridge University Press.

Fraley, C. and A.E. Raftery (2002), "Model-based clustering, discriminant analysis, and density estimation," *Journal of the American Statistical Association*, 97 (458), 611-31.

Ghose, A., P.G. Ipeirotis, and B. Li (2012), "Designing ranking systems for hotels on travel search engines by mining user-generated and crowdsourced content," *Marketing Science*, 31 (3), 493-520.

Ghose, S. and V.R. Rao (2007), "A choice model of bundles features and meta-attributes: An application to product design," *Working paper*.

Gnyawali, D.R. and B.-J.R. Park (2011), "Co-opetition between giants: Collaboration with competitors for technological innovation," *Research Policy*, 40 (5), 650-63.

Green, P.E. and F.J. Carmone (1969), "Multidimensional scaling: An introduction and comparison of nonmetric unfolding techniques," *Journal of Marketing Research*, 6 (3), 330-41.

Green, P.E., J.D. Carroll, and S.M. Goldberg (1981), "A general approach to product design optimization via conjoint analysis," *The Journal of Marketing*, 45 (3), 17-37.

Green, P.E. and V.R. Rao (1971), "Conjoint measurement for quantifying judgmental data," *Journal of Marketing Research*, 8 (3), 355-63.

Gutman, J. (1982), "A Means-End Chain Model Based on Consumer Categorization Processes," *Journal of Marketing*, 46 (2), 60-72.

Haley, R.I. (1968), "Benefit Segmentation: A Decision-Oriented Research Tool," *Journal of marketing*, 32 (3), 30-35.

Hansen, K. and V. Singh (2009), "Market structure across retail formats," *Marketing Science*, 28 (4), 656-73.

Hauser, J.R. and B. Wernerfelt (1990), "An evaluation cost model of consideration sets," *Journal of consumer research*, 16 (4), 393.

Hofstede, F.T., J.-B.E. Steenkamp, and M. Wedel (1999), "International Market Segmentation Based on Consumer--Product Relations," *Journal of Marketing Research*, 36 (1), 1-17.

Hu, M. and B. Liu (2004), "Mining and summarizing customer reviews," *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 168-77.

IDC (2011), "IDC Predicts 2012 Will Be the Year of Mobile and Cloud Platform Wars as IT Vendors Vie for Leadership While the Industry Redefines Itself," [available at http://www.idc.com/getdoc.jsp?containerId=prUS23177411].

---- (2012), "IDC Raises Tablet Forecast for 2012 and Beyond As iOS Picks Up Steam, Android Gains Traction, and Windows Finally Enters the Market [Press release]," [available at http://www.idc.com/getdoc.jsp?containerId=prUS23833612].

Johnson, M.D. (1988), "Comparability and Hierarchical Processing in Multialternative Choice," *Journal of Consumer Research*, 15 (3), 303-14.

Johnson, M.D. (1984), "Consumer choice strategies for comparing noncomparable alternatives," *Journal of Consumer Research*, 11 (3), 741-53.

Johnson, M.D. (1989), "On the nature of product attributes and attribute relationships," *Advances in Consumer Research*, 16 (5), 598-604.

Johnson, M.D., D.R. Lehmann, C. Fornell, and D.R. Horne (1992), "Attribute abstraction, feature-dimensionality, and the scaling of product similarities," *International Journal of Research in Marketing*, 9 (2), 131-47.

Kohli, R. and R. Krishnamurti (1989), "Optimal product design using conjoint analysis: Computational complexity and algorithms," *European Journal of Operational Research*, 40 (2), 186-95.

Koukova, N.T., P. Kannan, and A. Kirmani (2012), "Multiformat digital products: how design attributes interact with usage situations to determine choice," *Journal of Marketing Research*, 49 (1), 100-14.

LaValle, S., E. Lesser, R. Shockley, M.S. Hopkins, and N. Kruschwitz (2011), "Big data, analytics and the path from insights to value," *MIT Sloan Management Review*, 52 (2), 21-32.

Lee, T. and E. Bradlow (2011), "Automated marketing research using online customer reviews," *Journal of Marketing Research*, 48 (5), 881-94.

Lin, D. (1998), "Automatic retrieval and clustering of similar words," *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, 768-74.

Liu, B. (2012), *Sentiment Analysis and Opinion Mining*: Morgan & Claypool Publishers.

Luo, L., P. Kannan, and B.T. Ratchford (2008), "Incorporating subjective characteristics in product design and evaluations," *Journal of Marketing Research*, 45 (2), 182-94.

Maedche, A. and S. Staab (2002), "Measuring similarity between ontologies," in *Knowledge engineering and knowledge management: Ontologies and the semantic web*. New York: Springer.

---- (2004), "Ontology Learning," in *Handbook on Ontologies*, Steffen Staab and Rudi Studer, eds.: Springer Berlin Heidelberg.

Mayzlin, D. and J. Shin (2011), "Uninformative advertising as an invitation to search," *Marketing Science*, 30 (4), 666-85.

Myers, J.H. and E. Tauber (2011), *Market structure analysis*: Marketing Classics Press.

Nelson, P. (1999), "Multiattribute Utility Models," in *Consumer Research and Economic Psychology. Northampon, MA: Edward Elgar*, Peter E. Earl and Simon Kemp, eds.

Netzer, O., R. Feldman, J. Goldenberg, and M. Fresko (2012), "Mine your own business: Market-structure surveillance through text mining," *Marketing Science*, 31 (3), 521-43.

Netzer, O., O. Toubia, E.T. Bradlow, E. Dahan, T. Evgeniou, F.M. Feinberg, E.M. Feit, S.K. Hui, J. Johnson, and J.C. Liechty (2008), "Beyond conjoint analysis: Advances in preference measurement," *Marketing Letters*, 19 (3), 337-54.

Newman, J. (2013), "Small Windows 8 Tablets: Can You Spot the Differences?," [available at http://techland.time.com/2013/10/17/small-windows-8-tablets-can-you-spot-the-differences/].

Onishi, H. and P. Manchanda (2012), "Marketing activity, blogging and sales," *International Journal of Research in Marketing*, 29 (3), 221-34.

Pang, B. and L. Lee (2008), "Opinion Mining and Sentiment Analysis," in *Foundations and Trends in Information Retrieval*, Vol. 2: Now Publishers Inc.

Ratneshwar, S., A.D. Shocker, J. Cotte, and R.K. Srivastava (1999), "Product, person, and purpose: putting the consumer back into theories of dynamic market behaviour," *Journal of Strategic Marketing*, 7 (3), 191-208.

Sabou, M. (2005), "Learning web service ontologies: an automatic extraction method and its evaluation," *Ontology learning from text: methods, evaluation and applications*, 123.

Salton, G. and M.J. McGill (1983), "Introduction to modern information retrieval."

Sinofsky, S. (2012), "Building a rich and extensible media platform."

Sloane, G. (2011), "Out of Touch: HP exits tablet business," in The New York Post 08/20/2011, pp. 25.

Socher, R., B. Huval, C.D. Manning, and A.Y. Ng (2012), "Semantic compositionality through recursive matrix-vector spaces," *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 1201-11.

Socher, R., A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, and C. Potts (2013), "Recursive deep models for semantic compositionality over a sentiment treebank," *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1631-42.

Steel, E. (2013), "Americans spending more time on digital devices than TV screens," in Financial Times (London, England), pp. 1.

Steenkamp, J.B. and H. Van Trijp (1997), "Attribute elicitation in marketing research: a comparison of three procedures," *Marketing Letters*, 8 (2), 153-65.

Su, M. (2008), "Methods for Handling Massive Numbers of Attributes in Conjoint Analysis," in *Review of Marketing Research*, Naresh K. Malhotra, ed. Vol. 5: Emerald Group Publishing Limited.

Tirunillai, S. and G.J. Tellis (2012), "Does chatter really matter? Dynamics of user-generated content and stock performance," *Marketing Science*, 31 (2), 198-215.

Torres, A. and T.H. Bijmolt (2009), "Assessing brand image through communalities and asymmetries in brand-to-attribute and attribute-to-brand associations," *European Journal of Operational Research*, 195 (2), 628-40.

Tsuruoka, D. (2011), "HP May Exit The PC Business, Dumping Its TouchPad Tablet; Lowers Sales, Profit Outlook; CEO: Major makeover for No. 1 PC maker is all about transforming future," in Investor's Business Daily 08/18/2011, pp. A04.

Veksler, V.D., R.Z. Govostes, and W.D. Gray (2008), "Defining the dimensions of the human semantic space," *30th Annual Meeting of the Cognitive Science Society*, 1282-87.

Walsh, M. (2011), "Online Media Daily: Tablets Beat PCs In Media Consumption," (accessed 2/1, 2014), [available at http://www.mediapost.com/publications/article/152893/tablets-beat-pcs-in-media-consumption.html].

Yoffie, D.B. (1996), "Competing in the Age of Digital Convergence," *California Management Review*, 38 (4), 31-53.

Figures and Tables

Figure 1. Traditional Versus Text Analytics Approach to Market Structure Analysis

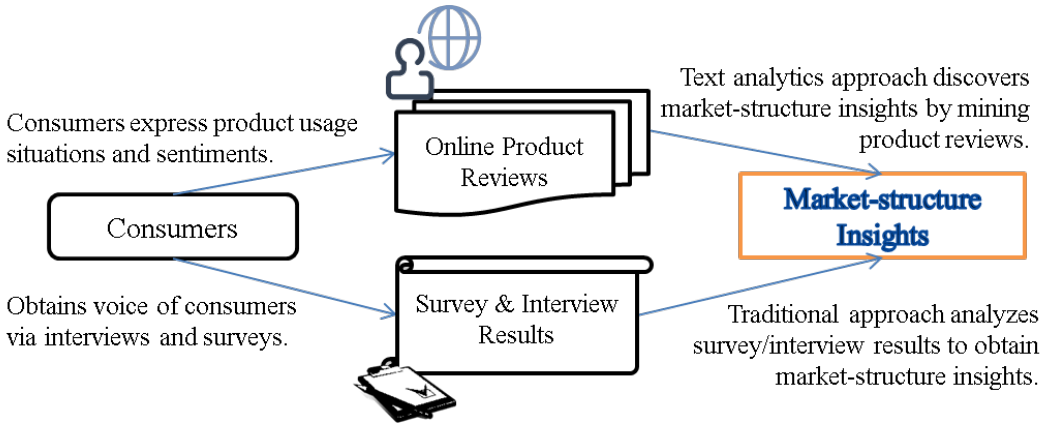


Figure 2. Semi-Structured and Unstructured Reviews

a. Reviews' Pros/Cons List

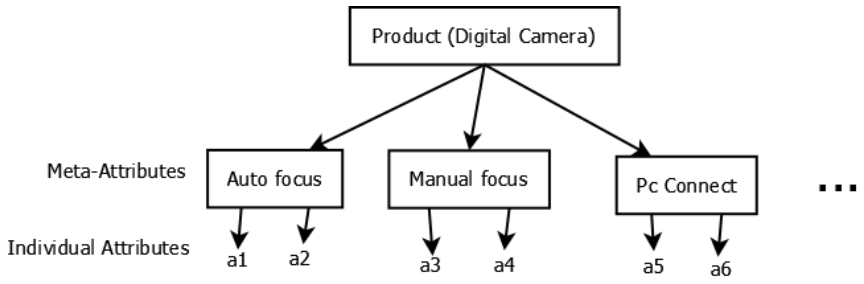
Pros: Strong Multi-tasking, sharp display
Cons: Case design, sluggish performance, many features difficult to find in the OS, lack of available apps.

b. Free-Form (Unstructured) Reviews

I was able to snag one of these while it was at a price of $150 through Amazon. It's a very good product when it comes to doing the basic stuff such as surfing the web. I use it for college purpose. I download a lot of word docs. in order to be able to read my papers when I'm on the go in school. I would prefer this touchpad more than the ipad because it feels more "liberating" to use it. It is not just a screen full of apps and quick clicks. The downfall on this product is that there aren't too many apps to use. The big plus in this is that you can do a lot of multitasking on it, unlike the ipad. If you're a college student, go for this simple high-tech device.

Figure 3. Meta-Attributes and the Hierarchical Structure of Product Attributes

a. Meta-Attributes Specified by Lee and Bradlow (2011)



b. Proposed Multilevel Attribute Hierarchy

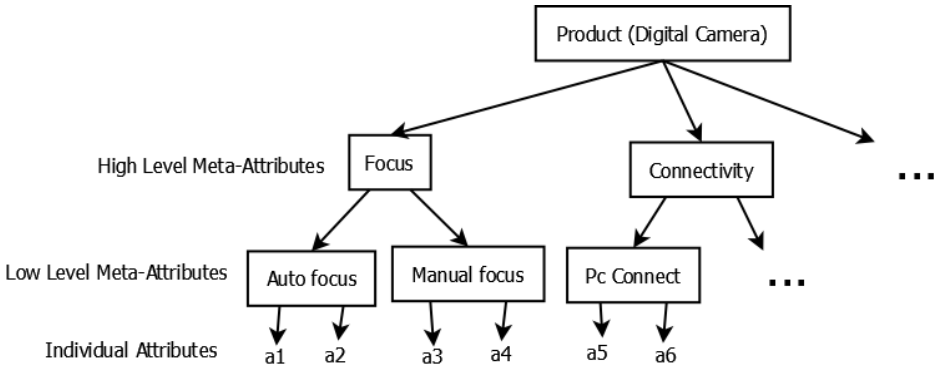


Figure 4. Text Analytics Method for Automatic Market Structure Analysis

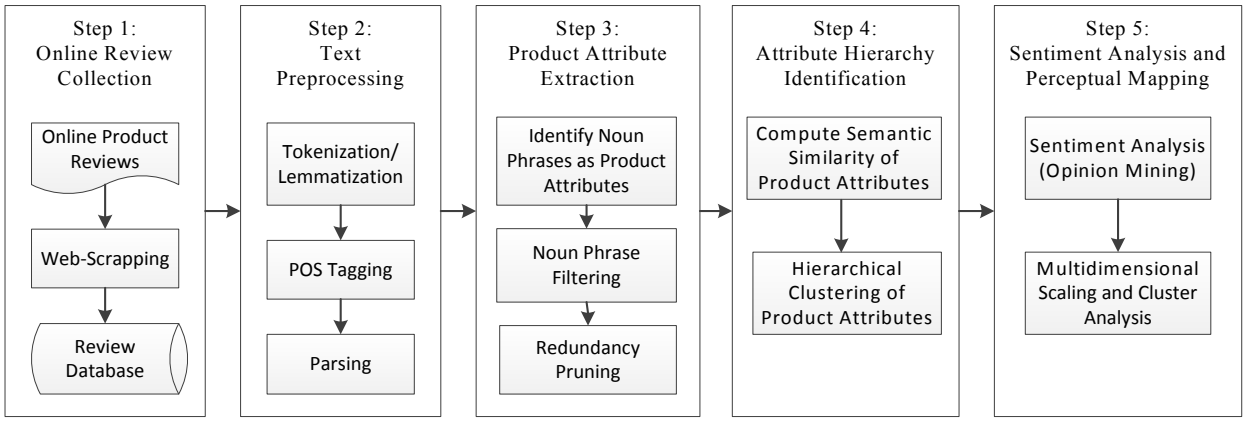


Figure 5. Dendrogram of Tablet's Attribute Hierarchy with Multilevel Meta-Attributes

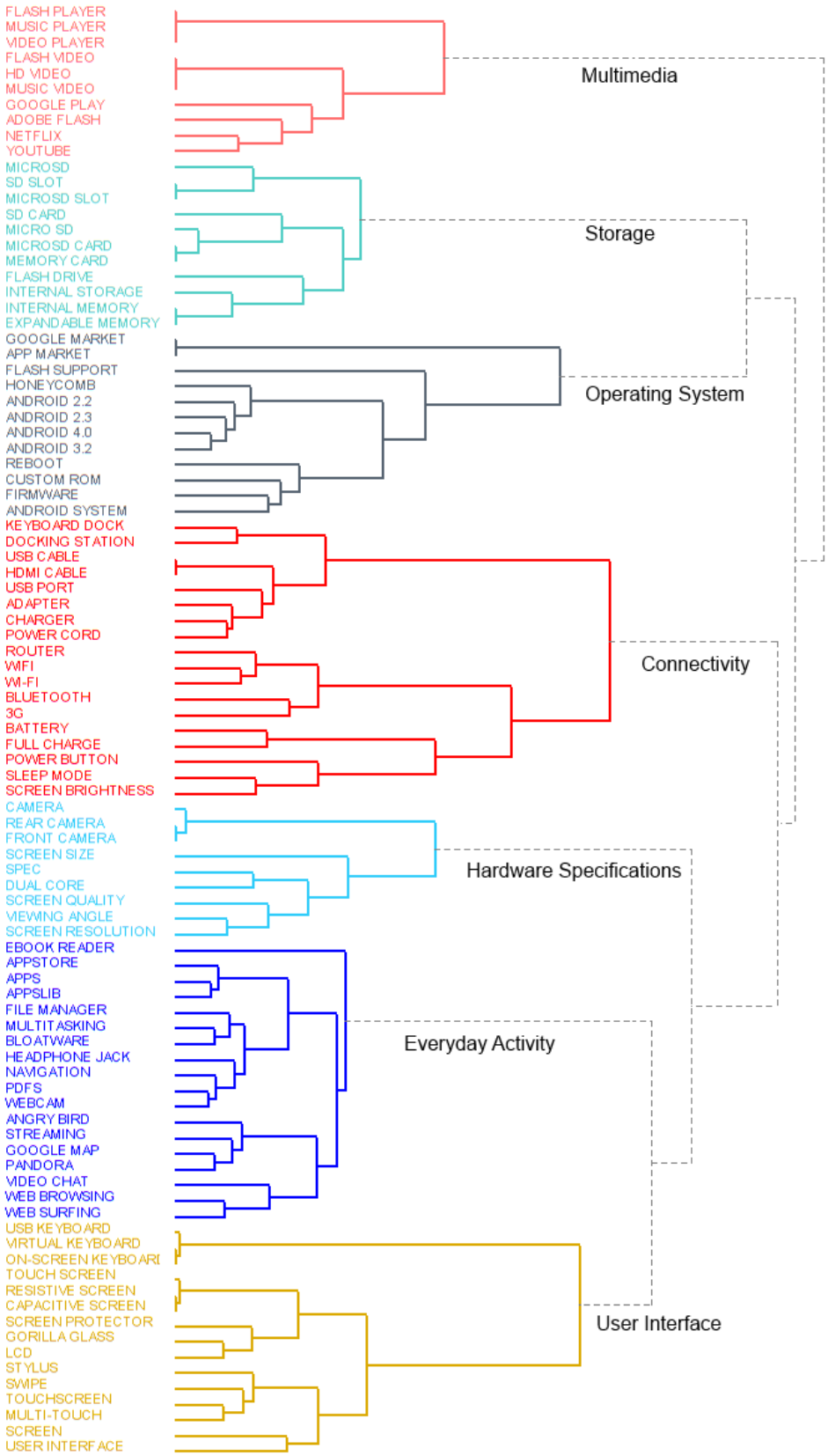


Figure 6. Hierarchical Structure of User Interface Meta-Attributes

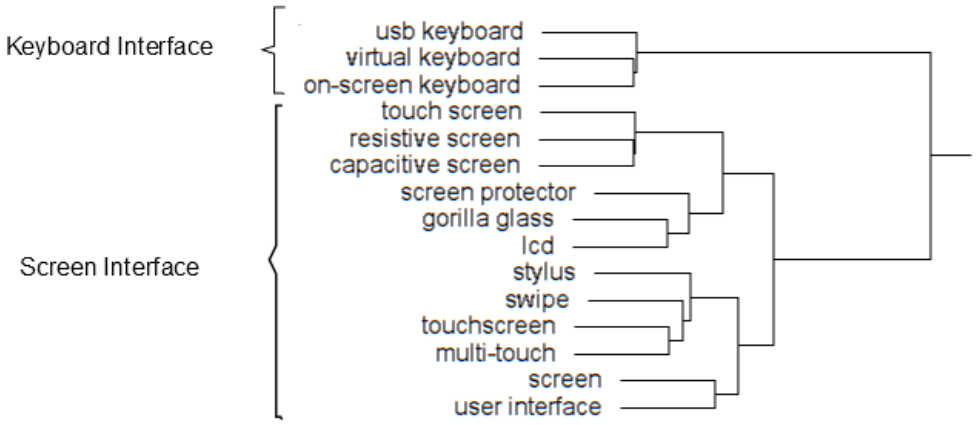


Figure 7. Sentiments for Six Major Tablet Brands

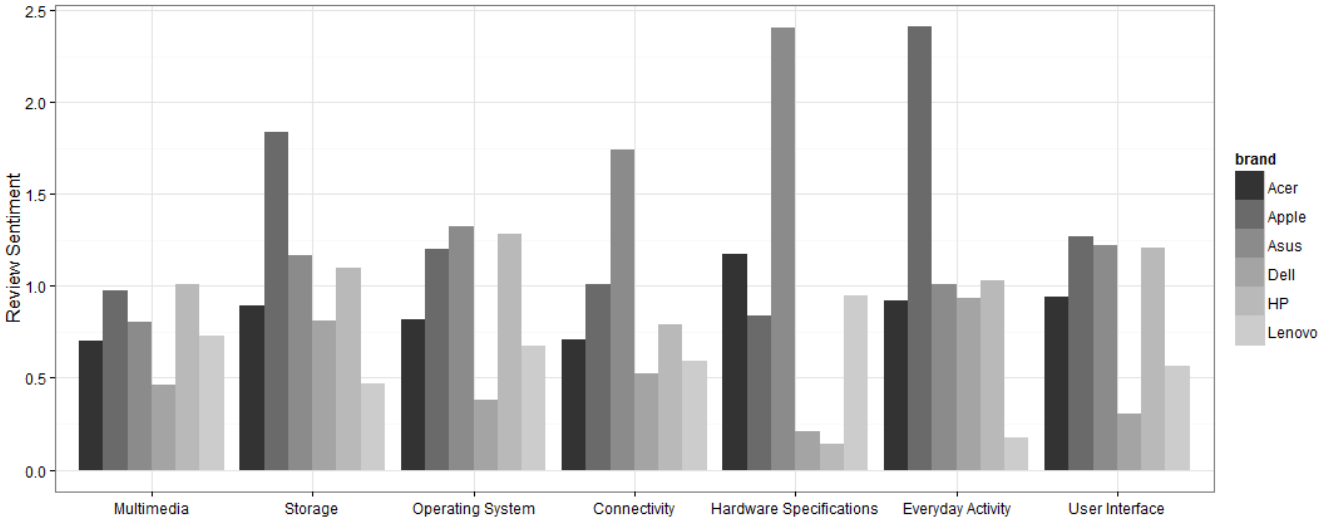


Figure 8. Multidimensional Scaling Performance

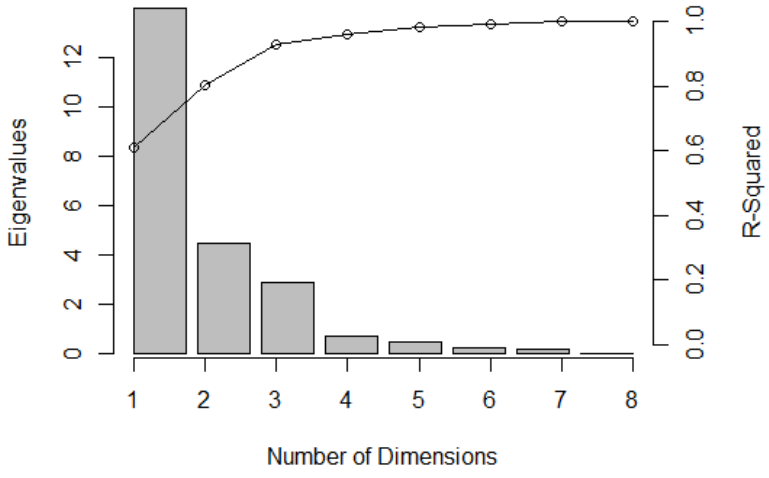


Figure 9. MDS Map for 15 Tablet Brands Based on Consumer Sentiments

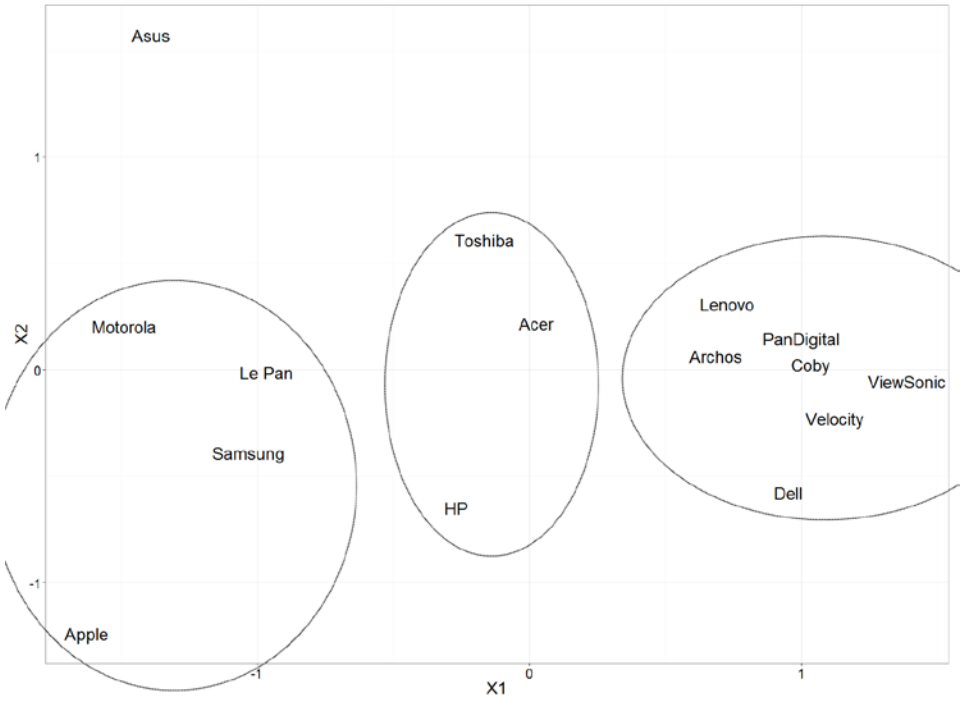


Figure 10. Market Structure Map for 15 Tablet Brands Based on Attribute Counts and CA

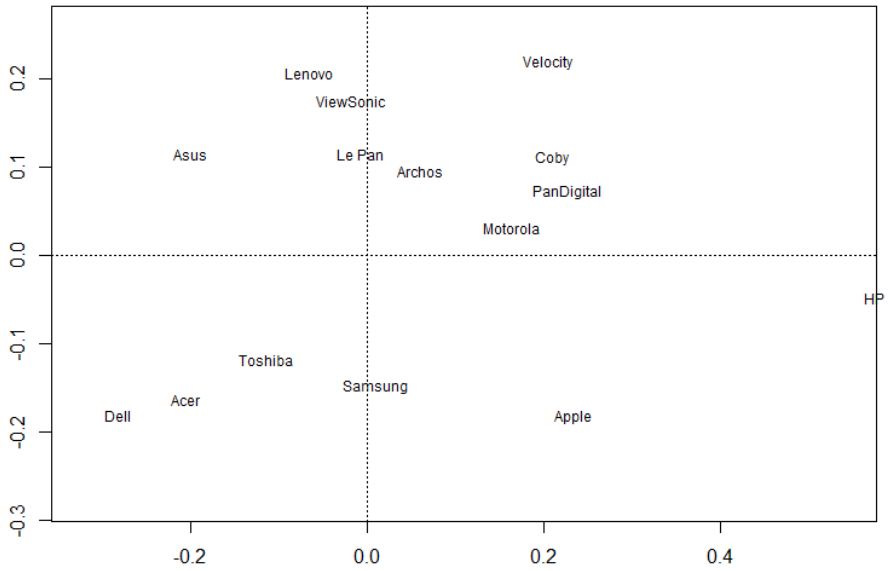


Figure 11. Market Structure Derived from Post-July 2012 Product Reviews

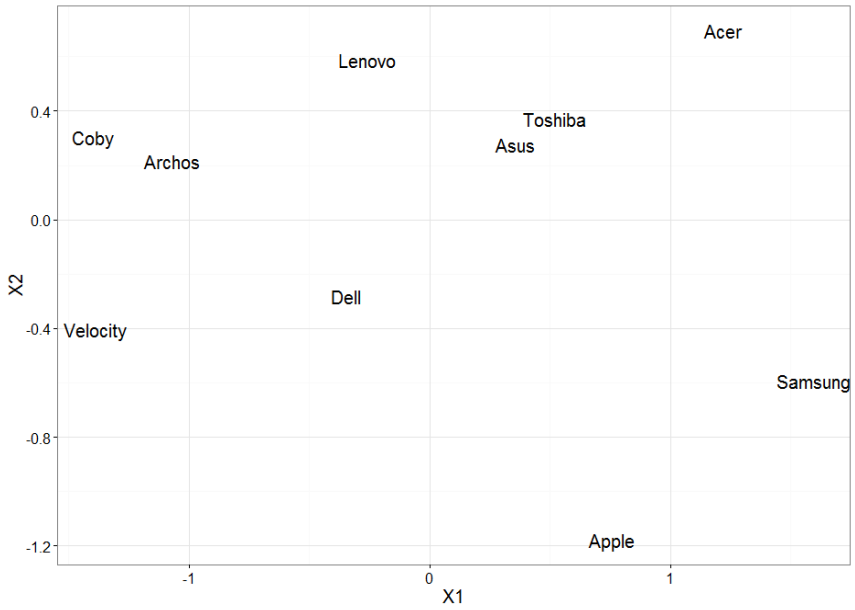


Figure 12. Correlation between Distance Matrices by Data Set Size

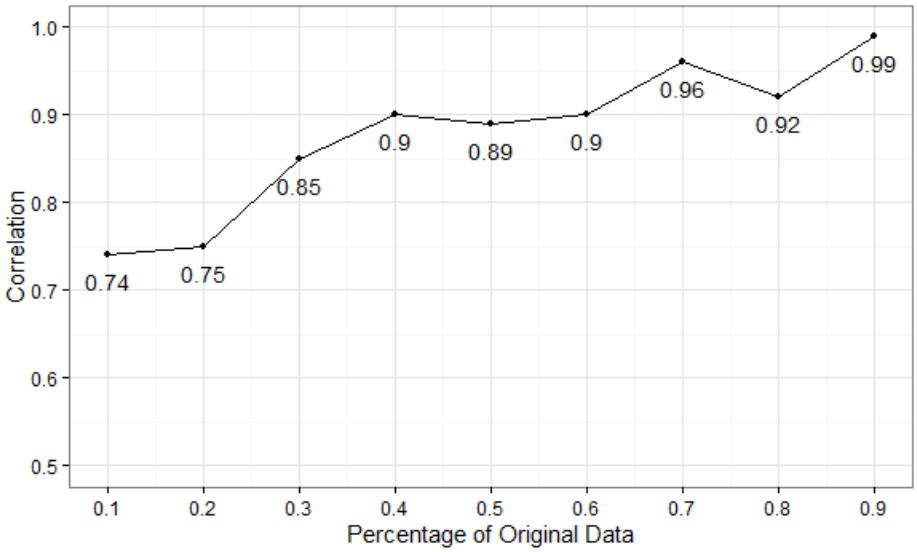


Table 1. Examples of Product Attribute Extraction and Filtering

Noun Phrases	Support	Pure Support	Pure Support/Support Ratio	Likelihood Ratio	Candidate Attribute?
Tablet	.459	.0881	.193	41053	No
One	.272	.0644	.236	20.2	No
Apps	.21	.0356	.169	12526	Yes
Time	.172	.0624	.3634	42.9	No
Battery Life	.1	.1	1	—	Yes
WiFi	.1	.034	.344	4948	Yes
Camera	.0792	.0295	.372	3263	Yes
Market	.0692	-.0398	-.5759	1748	No
Life	.02426	-.0757	-4.104	8259	No

Table 2. Tasks and Threshold Values for Product Attribute Extraction

Attribute Extraction Tasks	Threshold Value	Number of Phrases
Noun phrases extracted	—	137,028
Association rule mining	Support > .00385	1122
Redundancy pruning	Support ratio > .1	761
Likelihood ratio filtering	LR > 2000	231
Number of attributes identified	—	93

Table 3. Semantic Vectors and Similarity Measures for Product Attributes

a. Semantic Vectors with PMI Scores

YouTube		Webcam		USB Cable	
Usage Situation	PMI Score	Usage Situation	PMI Score	Usage Situation	PMI Score
[dobj, watch]	4.1809	[dobj, face]	4.9863	[dobj, charge]	3.4966
[dobj, play]	2.8208	[nsubj, glossy]	4.5754	[dobj, recharge]	3.4165
[nsubj, glossy]	2.4220	[nsubj, flicker]	4.2510	[dobj, connect]	3.3397
[nsubj, show]	2.1428	[nsubj, vivid]	4.0638	[dobj, include]	3.1354
[nsubj, flicker]	2.0975	[nsubj, crisp]	4.0458	[dobj, recognize]	2.6499
[dobj, download]	1.8577	[nsubj, gorgeous]	3.9533	[dobj, break]	2.2564
[nsubj, slow]	1.2321	[nsubj, brighter]	3.8663	[dobj, insert]	2.1343
[nsubj, resolution]	1.2075	[nsubj, dark]	3.7915	[dobj, require]	2.1106

Notes: dobj = direct object, nsubj = nominal subject.

b. Cosine Similarity

	USB Cable	Webcam	YouTube
USB Cable	1	—	—
Webcam	.1938	1	—
YouTube	.3263	.4071	1

Table 4. Common Usage Situations of Seven Meta-Attributes

Meta-Attributes	**Common Usage Situations**
Multimedia	watch, view, download, play
Storage	add, insert, recognize, provide
Operating System	update, install, load, upgrade
Connectivity	charge, drain, connect, access, remove, recognize
Hardware Specification	compare, test, improve
Everyday Activity	play, download, install, run, work, maintain, enjoy
User Interface	touch, see, find, rotate, scratch, protect, calibrate

Table 5. Comparison of Text-Mined Product Attributes with Expert Guides

a. Discovered Attributes versus Attributes Extracted from Expert Guides

Our Results	**Eopinions.com**	***Consumer Reports***	**Amazon**	**eBay**
Multimedia	Audio output, Audio input			
Storage			Storage	Storage
Operation System	Platform, OS	OS	OS	OS
Connectivity	Network type, Wireless capabilities	Wireless connectivity, USB ports	Connectivity	Keyboard accessories
Hardware Specification		Screen size & shape, Display	Screen Size	Screen size, Processor type
Everyday Activity	Supported file types			
User Interface	Input method, Display tech			
		Printing capability		

b. Precision and Recall Compared with Expert Guides

	Eopinions.com	*Consumer Reports*	Amazon	eBay
Precision (P)	.71	.43	.57	.57
Recall (R)	1	.75	1	1

Table 6: Empirical Evaluation Results of Seven Meta-Attributes

Meta-Attributes	**Similarity with Attributes in Cluster**	**Similarity with Attributes outside Cluster**
Multimedia	1	.43
Storage	1	.63
Operating System	.57	.40
Connectivity	.82	.50
Hardware Specification	.89	.46
Everyday Activity	.77	.52
User Interface	.71	.40

Table 7. Top 15 Tablet Brands

Brand Name	Product	OS	Country	Main Product	RAM (GB)	CPU (GHz)	Storage (GB)	Battery (Hours)	Screen (in)	Screen Width (Pixel)	Screen Height (Pixel)	Weight (lb)	Number of SKUs in Study
Acer	Iconia	Android	Taiwan	PC	1	1	16	8	10.1	1280	800	1.6	20
Apple	iPad	IOS	USA	PC/Phone	.512	1	32	10	9.7	1024	768	1.4	44
Archos	Home tablet	Android	French	Media Player	.512	1	8	7	8	1024	600	1	29
Asus	Transformer	Android	Taiwan	PC	1	1.3	32	8	10.1	1280	800	2.1	27
Coby	Kyros	Android	USA	TV/DVD	.512	1	4	6	7	800	480	.87	33
Dell	Streak, convertible laptops	Android, Windows	USA	PC	1.5	1.25	168	4	8.55	1083	624	1	8
HP	TouchPad, convertible laptops	WebOS, Windows	USA	PC	3.5	1.86	160	5	12.1	1280	800	3.9	18
Le Pan	TC 970	Android	Japan	Tablet	.756	1.1	5	6.5	9.7	1024	768	1.47	2
Lenovo	ThinkPad, convertible laptops	Android, Windows	China	PC	1	1.6	120	7	10.1	1024	768	3.31	19
Motorola	Xoom	Android	USA	Phone	1	1	32	10	10.1	1280	800	1.6	4
PanDigital		Android	USA	Picture frame	.256	.8	2	6	7	800	600	1.1	13
Samsung	Galaxy	Android	South Korea	PC/Phone	1	1	16	9	8.9	1280	768	1.18	35
Toshiba	Thrive	Android	Japan	PC	1	1.2	16	8.5	10.1	1280	800	1.6	15
Velocity	Cruz	Android	USA	PC	.384	.66	3	8	7	800	600	1	10
ViewSonic	G Tablet	Android	USA	TV	.512	1	4	6	10	800	600	1.9	15

Table 8. Comparison of Consumer Sentiments for Lenovo and Budget Brands

	95% CI for Lenovo – Budget Brands Sentiments	
Meta-Attributes	Pre-July 2012	Post-July 2012
Multimedia	(.15, .82)*	(.17, .50)*
Storage	(-.29, .49)	(.21, .90)*
Operating System	(-.21, .48)	(.07, .49)*
Connectivity	(-.29, 2.33)	(.06, 1.74)*
Hardware Specification	(-.53, 7.60)	(.70, 4.91)*
Everyday Activity	(-.90, -.09)*	(-.02, 1.80)
User Interface	(-.76, 1.26)	(.66, 3.43)*

*Statistically significant at the 5 percent alpha level.

Table 9. $\overline{\text{TO}}$ Measure between Hierarchical Clustering and *k*-Medoids

Meta-Attributes	$\overline{\text{TO}}$ **Measure**
Multimedia	.92
Storage	.92
Operating System	.80
Connectivity	.50
Hardware Specification	.54
Everyday Activity	.35
User Interface	.52

Table 10. Comparison of Lexicon Approach and Sentiment Classification Approach

	Positive				**Negative**			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Lexicon Approach	.692	.636	.687	.661	.700	.731	.384	.504
SVM	.766	.741	.685	.712	.781	.727	.615	.666

Table 11. Comparison of Sentiment MDS and Attribute Count CA for Market Structure Analysis

		Correlation	
		MDS with Consumer Sentiment	CA with Attribute Counts
Factiva news	Lift	.603	.459
	PMI	.589	.407
	Cosine	.592	.458
	NGD	.633	.450
Brand co-mention	Lift	.714	.437
	PMI	.689	.338
	Cosine	.652	.409
	NGD	.612	.367

Table 12. Comparison of Market Structure Analysis Methods

	Traditional Methods	**Lee and Bradlow (2011)**	**Netzer et al. (2012)**	**Our Method**
Data Acquisition				
Source	Surveys	Pros/cons product reviews	Free-form discussion on web forum	Free-form online product reviews
Cost	High	Low	Low	Low
Analysis Methods				
Similarity between brands		Attribute counts	Co-mention of brands	Product attributes
Linguistic structures of sentences	Yes (manually)	No	No	Yes
Perceptual map	MDS	CA	MDS	MDS
Consumer sentiments	Yes	Through human tagged pros/cons labels	Common problems	Through sentiment analysis using opinion mining techniques
Implementation and Results				
Product attribute extraction	Elicitation procedures relying on human experts	Automatic discovery	No	Automatic discovery
Attribute structure	Yes, vary	Single level	N/A	Hierarchical structure
Product usage	Surveys	No	Common problems	Keywords of usage situations
Data sources for validation		Survey and consumer report	Transaction-based data	Survey, consumer report, news, proxy to transaction-based data

APPENDIX

A. Automatic Market Structure Analysis: Techniques, Implementation, and Empirical Study

A1. Collection and Preprocessing of Tablet Product Reviews

Our method collects online product reviews automatically for a product category of interest, such as tablet computers, as used in the empirical study. The Java-based web crawler we wrote downloaded product reviews under the *Tablets & Tablet PCs* category from Amazon.com and cleaned the HTML tags with the jsoup library (Hedley 2010). In addition to the free-form text review, we downloaded star ratings, the date of the review, and general product information such as brand and product name. This information was stored in a SQLite database.

During preprocessing, we corrected commonly misspelled words and adjusted end-of-sentence symbols that could affect the accuracy and implementation of downstream text analysis and mining tasks (Subramaniam et al. 2009). Spelling corrections substituted terms from a precompiled list (available at wordpress.org). The end-of-sentence symbol adjustment used regular expression substitution. For example, many reviewers used ellipses instead of a single period between sentences, but parsers often assume that ellipses appear only in the middle of a sentence. To help the part-of-speech (POS) parser determine sentence breaks, we replaced one or more successive periods with a single period. The preprocessing procedure also identified product reviews' linguistic components for further text analysis, with three steps:

1. Tokenization, to separate words and detect sentence boundaries. It breaks reviews into sequences of elementary units, such as individual words, sentences, and punctuation marks. Simultaneous lemmatization transforms words to their root forms and removes variance; for example, *computers* and *computer's* become *computer*, and *are, am,* and *is* become *be*.

2. Part-of-speech (POS) tagging, to identify words by their parts of speech, such as adjectives, nouns, noun phrases, verbs, or verb phrases. We used the Penn Treebank Tags (Marcus et al. 1993) for POS representation. For example, the sentence, "The handwriting recognition is fantastic," would be tagged: "The (DT) handwriting (NN) recognition (NN) is (VBZ) fantastic (JJ)," where NN stands for a noun, VBZ indicates a third-person singular verb, and JJ is an adjective.
3. Dependency parsing, to infer grammatical relationships between words. For example, the sentence, "The handwriting recognition is fantastic," produces the following dependency pairs: "The" as a determinant of "recognition," "handwriting" as a noun compound modifying "recognition," and "recognition" as nominal subject of "fantastic." A complete list of the grammar dependencies is available from De Marneffe and Manning (2008a).

Several open-sourced natural language processing (NLP) packages are available to handle these preprocessing tasks. We chose the Stanford CoreNLP (De Marneffe and Manning 2008b; Klein and Manning 2003), distributed freely under the GNU General Public License.

A2. Product Attribute Extraction

Integrating the NLP techniques enabled us to infer product attributes. Previous researchers have investigated various techniques to elicit product attributes automatically. Unsupervised learning approaches are preferable (Wei et al. 2009), because they do not require annotated review sentences for training purposes. The framework proposed by Hu and Liu (2004), which employs association rule mining (Agrawal and Srikant 1994) to identify frequent noun phrases as opinion features from reviews, has proven valid (Archak et al. 2011). We expand on their framework by integrating association rule mining with a sequence of filtering and pruning techniques to extract noun phrases from product reviews as candidate product attributes. Association rule mining often serves to analyze sales data, including transactions of

market baskets (co-purchased items). To apply association rule mining, each set of noun phrases is treated as a set of items *I*, and each review is a transaction that consists of a subset of items. An association rule mining algorithm generates frequent item sets, with support greater than a specified level.

Our method starts by extracting noun phrases that include fewer than some predefined number of words (e.g., 3) from review sentences. Stop words such as *I*, *the*, *was*, and *a* are filtered out of the noun phrases. The most frequently mentioned noun phrases become candidates for product attributes. Several measures are available to resolve three key problems with extracting product attributes automatically and effectively:

1. Common noun phrases that are not specific to the product category of interest (e.g., tablets), such as *something*, *people*, *fact*, *others*, or *today*.
2. Redundancy nouns that are parts of other frequent noun phrases, such that *life* is a redundant noun with respect to the noun phrase *battery life*.
3. Noun phrases that are brand names and general product categories, such as *iPad*, *Samsung*, *tablet*, and *tablet computer*.

The first problem requires filtering out common and irrelevant noun phrases. To do so, we applied a likelihood ratio test (Yi et al. 2003). For each noun phrase identified, our method computed relative frequency discrepancies between reviews of the product category of interest (e.g., tablets) and reviews of an irrelevant product category (e.g., books). We used reviews from a different product category, instead of a more general corpus, because many noun phrases are specific to e-commerce (e.g., *shipping, Amazon*) but do not refer to the product, so they should not display any substantial frequency discrepancy in the reference model. Noun phrases with high likelihood ratios are candidate product attributes for the product category of interest. Phrases with likelihood ratios below a threshold are considered irrelevant and can be eliminated.

For example, “touchscreen” should appear frequently in tablet reviews but not in book reviews, so it would produce a high likelihood ratio.

Mathematically, the likelihood ratio $-2\log\lambda$ is defined as:

$$-2\log\lambda = \begin{cases} -2\ lr & if\ r_2 < r_1 \\ 0 & if\ r_2 \geq r_1 \end{cases},$$

where $lr = (C_{11} + C_{21})\log r + (C_{12} + C_{22})\log(1-r) - C_{11}\log r_1 - C_{12}\log(1-r_1) - C_{21}\log r_2 - C_{22}\log(1-r_2)$,

$$r_1 = \frac{C_{11}}{C_{11}+C_{12}}, r_2 = \frac{C_{21}}{C_{21}+C_{22}}, r = \frac{C_{11}+C_{21}}{C_{11}+C_{12}+C_{21}+C_{22}},$$

and C_{11}, C_{12} are counts of relevant product reviews that contain and do not contain the phrase, respectively, whereas C_{21}, C_{22} are counts of irrelevant texts that contain and do not contain the phrase, respectively. Although the likelihood ratio is asymptotically χ^2 distributed, in practice the filtering threshold should be set much higher than a theoretical $p = .05$ level.

Our method adopts the redundancy pruning procedure proposed by Hu and Liu (2004) to solve the second problem, which computes a measure of how often a phrase appears alone, rather than as part of another phrase. When applying association rule mining, the estimated percentage of reviews that contain a phrase constitute support for the phrase, and the measure of a phrase appearing alone is defined as *pure support*. Consider an example: Pure support for the term *life* is the percentage of reviews that contain *life* as a noun phrase but no supersets (e.g., *battery life*) of that phrase. When a phrase’s pure support is lower than its support, the phrase by itself carries less meaning in that context. Therefore, we retain only noun phrases with a pure support-to-support ratio greater than a threshold. Although in Hu and Liu's (2004) original definition, pure support cannot be negative, we employed a heuristic to achieve faster computation speed that

may result in negative values for pure support. This variation should not affect the results in practice.

For the third problem, we manually compiled a set of unrelated noun phrases, such as brand and product names, and removed them from the list. Figure 1 presents the pseudo-code for the product attribute extraction.

Figure 1. Product Attribute Extraction Algorithm

Algorithm 1 Attribute Extraction

Input: R: set of reviews for relevant products; IR: set of reviews for irelevant products
Output: C: set of product attributes for relevant products
Parameters: lr: likelihood ratio threshold; s: support threshold; ps: pure support/support threshold; s_0: initial support cutoff ($s_0 \ll s$)

procedure ASSOCIATION RULE MINING
  for each $review_i \in R$ **do**
    Transaction $t_i \leftarrow$ Noun phrases (NPs) in $review_i$
    Add NPs in review i to item set I
  end for
  Calculate Support for all NPs in I
  Candidate NPs $C \leftarrow \{\text{NP}_j \in I | Support(\text{NP}_j) \geq s_0\}$
end procedure

procedure CALCULATE PURE SUPPORT
  for each $NP_j \in C$ **do**
    $PureSupport(NP_j) \leftarrow Support(NP_j)$
    for each $NP_k \in C$ s.t. $i \neq k$ **do**
      if NP_j is a subset of NP_k **then**
        $PureSupport(NP_j) \leftarrow PureSupport(NP_j) - Support(NP_j)$
      end if
    end for
  end for
end procedure

procedure CALCULATE LIKELIHOOD RATIO
  for each $review_i \in IR$ **do**
    Transaction $t_i \leftarrow$ Noun phrases (NPs) in $review_i$
    Add NPs in review i to item set I_{ir}
  end for
  Calculate $Support_{ir}$ for all NPs in I_{ir}
  for each $NP_j \in C \cap I_{ir}$ **do**
    $C_{11}^j \leftarrow Support(NP_j) \times |R|, C_{12}^j \leftarrow |R| - C_{11}^j,$
    $C_{21}^j \leftarrow Support_{ir}(NP_j) \times |IR|, C_{22}^j \leftarrow |IR| - C_{22}^j,$
    $r_1^j \leftarrow \frac{C_{11}^j}{C_{11}^j + C_{12}^j}, r_2^j \leftarrow \frac{C_{21}^j}{C_{21}^j + C_{22}^j}, r^j \leftarrow \frac{C_{11}^j + C_{21}^j}{C_{11}^j + C_{12}^j + C_{21}^j + C_{22}^j}$
    if $Support_{ir} > Support_r$ **then**
      $LR(NP_j) \leftarrow 0$
    else
      $LR(NP_j) \leftarrow -2[(C_{11}^j + C_{21}^j)\log r^j + (C_{12}^j + C_{22}^j)\log(1 - r^j) - C_{11}^j \log r_1^j - C_{12}^j \log(1 - r_1^j) - C_{21}^j \log r_2^j - C_{22}^j \log 1 - r_2^j]$
    end if
  end for
end procedure

procedure FILTERING
  for each $NP_j \in C$ **do**
    if $LR(NP_j) \leq lr$ **or** $Support(NP_j) \leq s$ **or** $PureSupport(NP_j) \leq d$ **then**
      Remove NP_j from C
    end if
  end for
end procedure

A3. Attribute hierarchy identification

Our method identifies a multilevel hierarchical structure for product attributes by measuring semantic similarities among attributes. We propose a new measure of semantic similarity by combining marketing theories with a classic distributional similarity (Lin 1998a). We first infer consumers' usage situations by analyzing grammatical relationships (i.e., verb phases associated with product attributes). For example, in the sentence, "The USB supports keyboards," the subject USB performs the action of *support* on the direct object keyboards, which reveals the usage situation. For each product attribute, we quantitatively summarize all specific grammatical relationships expressed in the review with a semantic vector. The similarity measure between two attributes is the number of usage situations they share, so similarity is measured as the cosine distance between two attributes' semantic vectors.

We represent each product attribute's usage situations with a semantic vector of the same dimension. A dependency parser applies to extract dependency relationships related to product attributes. Within a sentence, dependency relationships appear as grammatical relationships among words. By using the dependency parser rather than a POS tagger (Archak et al. 2011), we identify complex linguistic structures, even if the related word is distant from the particular product attribute. Our method employs the seven dependency relationships in Table 1 (De Marneffe and Manning 2008b) to capture usage situations.

We generate a semantic vector for each product attribute that collects the dependency counts across reviews. Every entry in the semantic vector consists of three parts: type of dependency, the associated word, and a frequency count. If the product attribute *picture* serves as the direct object (*dobj*) of the verb *enlarge* 10 times in the entire collection of reviews and as the

direct object of the verb *browse* 2 times, in the semantic vector for the product attribute "picture," we include two corresponding entries: $[dobj, enlarge, 10]$ and $[dobj, browse, 2]$.

We calculate frequency counts for the extracted product attributes, then construct the semantic vectors with the same list of [*dependency relationship, word*] combinations to compute semantic similarity among product attributes. If a product attribute lacks a certain [*dependency relationship, word*] combination, the frequency count is 0. For our empirical study, the dimensions of initial semantic vectors, which include 93 product attribute [*dependency relationship, word*] combinations, exceeded 50,000. Because the quality of similarity scores can be limited by large vector dimensions, we followed a common dimension reduction practice and set the minimal frequency of the dependency count to 10 (Geffet and Dagan 2004), which drastically reduced the dimensions to 264.

Table 1: Seven Dependency Relationships to Infer Usage Situations for Product Attributes

Dependency Relationship	**Description**
Relative clause modifier	A relative modifier of a noun phrase (NP) is a relative clause modifying the NP. The relation points from the head noun of the NP to the head of the relative clause, normally a verb.
Direct object	The direct object of a verb phrase (VP) is the noun phrase, which is the (accusative) object of the verb.
Indirect object	The indirect object of a VP is the noun phrase that is the (dative) object of the verb.
Nominal subject	A nominal subject is a noun phrase that is the syntactic subject of a clause.
Controlling subject	A controlling subject is the relation between the head of an open clausal complement and the external subject of that clause.
Clausal complement	A clausal complement of a verb or adjective is a dependent clause with an internal subject that functions like an object of the verb or adjective.
Prepositional modifier	A prepositional modifier of a verb, adjective, or noun is any prepositional phrase that serves to modify the meaning of the verb, adjective, noun, or even another preposition.

In the final calculation, we replace the raw count of dependence frequencies with the pointwise mutual information (PMI) (Turney 2001) between an attribute and its dependence words. The purpose is to reduce the impact of common words that carry little information, such as "have" or "use." In co-occurrence semantic vectors using PMI scores, these general verbs

offer little value for distinguishing specific usage situations and take less weight; entries with tighter associations with product attributes take more weight. The PMI of two words, $word_1$ and $word_2$, is:

$$\text{PMI}(word_1, word_2) = \log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)},$$

where $P(word_1, word_2)$ is the number or probability of co-occurrences in a certain window (e.g., sentence). Although PMI works well for word-context matrices (Pantel and Lin 2002), it can be biased toward infrequent events (Turney and Pantel 2010). Laplace smoothing corrects this effect by adding a constant value to raw frequencies of word occurrence and co-occurrence counts (Turney and Littman 2003). For example, add-one smoothing performs the following correction:

$$\text{PMI}(word_1, word_2) = \log_2 \frac{C(word_1, word_2)+1}{(C(word_1)+1)(C(word_2)+1)},$$

where $C(word_1)$ is the frequency of sentences that contain $word_1$.

This definition of semantic vectors can result in vectors with large dimensions, because any word that shares a dependency relationship with a product attribute is a distributional entry. To avoid the curse of dimensionality, we might include only entries with frequency counts or PMI values above a certain threshold (Geffet and Dagan 2004). The thresholds can be set initially according to the size of product reviews for market structure analysis and subsequently adjusted.

Using a semantic vector to represent each attribute's usage situations, we next calculate the semantic similarity between each pair of attributes to capture their shared usage situations. The most popular measure is cosine similarity (Turney and Pantel 2010). If x and y are two semantic vectors with the same dimension, the cosine of the angle between them is:

$$\cos(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}.\boldsymbol{y}}{\|x\|\|y\|} = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2 \sum y_i^2}} \ .$$

We use cosine distance instead of the more common Euclidean norm for several reasons. If x and y are vectors that describe the usage situations of two product attributes, and the attribute represented by x appears fewer times than the attribute represented by y in reviews, x will have a smaller Euclidean length than y, even if both product attributes provide similar functional benefits. By using the cosine distance, we make the Euclidean length of a semantic vector irrelevant in the measure.

Lin (1998b) shows that semantic similarity based on a large collection of text conforms with the definition of similarity in information theory. Therefore, we can analytically assess relationships among product attributes and use the attribute similarity matrix for our subsequent cluster analysis to construct an attribute hierarchy with multiple levels of meta-attributes.

In addition, we use hierarchical cluster analysis to construct the attribute hierarchy. Our method follows Punj and Stewart's (1983) suggestion to address four issues: data transformations, solution, validity, and variable selection. We have already discussed the data transformation and variable selection; we represent each product attribute with a semantic vector of dependency tuples that describe usage situations. To cluster attributes, we apply the hierarchical clustering procedure with Ward's minimum variance linkage, which produces a nested sequence of partitions with an all-inclusive cluster at the top and individual product attributes at the bottom. Between the top and bottom levels range multiple levels of meta-attributes. Unlike *k*-means or *k*-centroid clustering, hierarchical clustering does not assume a particular number of clusters a priori.

A dendrogram graphically presents the hierarchical order in which product attributes are aggregated. We can cut through the dendrogram at a particular level to obtain the required

number of meta-attributes. The dendrogram is a representation of the attribute hierarchy, used to not only describe the consumer usage situations and preferences of a product category but also define the domain ontology of a product category in terms of its attributes and their aggregation relationships. Figure 2 provides the pseudo-code of attribute hierarchy identification.

Figure 2. Attribute Hierarchy Identification Algorithm

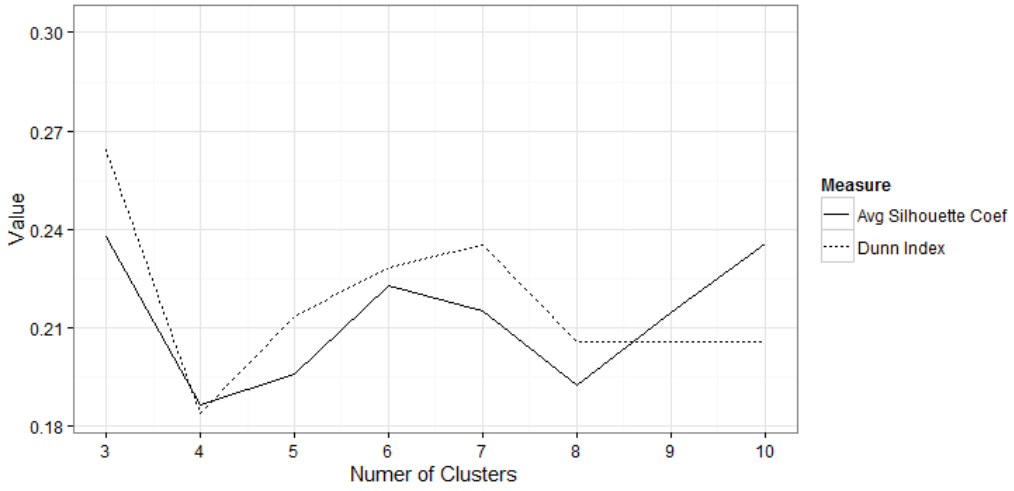
Algorithm 2 Building Hierarchical Structure of Product Attributes

Input: R: set of review sentences; A: set of product attributes
Output: C: dendrogram of product attributes
Parameters: DR: set of dependency relationships indicating usage situations;
Initialize empty set P to store related phrases
for each $sentence_i \in R$ **do**
  Parse dependency relationships
  for each dependency relationships $[type, phrase_1, phrase_2]$ **do**
    if type $\in DR$ **and** ($phrase_1$ **or** $phrase_2 \in A$) **then**
      if phrase2 $\in A$ **then**
        Semantic vector $attribute[type, phrase_1, count] \leftarrow attribute[type, phrase_1, count + 1]$
        Add $phrase_1$ to P
      else
        Semantic vector $attribute[type, phrase_2, count] \leftarrow attribute[type, phrase_2, count + 1]$
        Add $phrase_2$ to P
      end if
    end if
  end for
end for
Cut down number of entries in semantic vectors by count threshold if necessary
for each product attribute $a_i \in A$ **do**
  for each phrase $p_j \in P$ **do**
    Calculate $PMI(a_i, p_j)$
  end for
  For all entries in semantic vector $a_i[type, phrase, count] \leftarrow a_i[type, phrase, PMI(phrase, a_i)]$
end for
for each product attribute $a_i, a_j \in A$ **do**
  Calculate cosine similarity $\cos(a_i, a_j)$
end for
Conduct hierarchical clustering of atrribute set A based on cosine similarity

To examine the validity of the clustering results, we first examine qualitatively whether the results are meaningful and useful (Punj and Stewart 1983). As we show in the manuscript, the meta-attributes represented by the clusters reflect meaningful differentiations in usage situations. Quantitatively, Figure 3 demonstrates that according to the internal validation criteria—namely, the average Silhouette coefficient and Dunn index—the seven-cluster scenario (i.e., seven meta-attributes) is reasonable for our empirical study. The average Silhouette coefficient (Rousseeuw 1987) combines measures of both cohesion and separation for observations in a cluster, defined as $\frac{(b_i - a_i)}{\max(a_i, b_i)}$, where a_t is the average distance from observation t

to the other points in its cluster, and b_t is the minimum average distance from t to the clusters that do not contain this observation. The Dunn index (Dunn 1974) is another measure to identify compact, well-separated clusters, using the ratio of the smallest distance between observations from different clusters to the largest distance between observations from the same cluster. For both measures, greater values are desirable. According to Figure 3, when the number of clusters is 7, both measures are relatively high.

Figure 3. Internal Measures of Cluster Validity



A4. Sentiment Analysis

The sentiment analysis reveals consumers' sentiment polarities toward product attributes. A set of 2000 review sentences was chosen randomly and manually tagged by two coders into three sentiment polarities: positive, negative, or neutral. The tagged set served to train two binary sentiment classifiers using machine learning algorithms, one for detecting positive and one for detecting negative opinions. This approach is known as the "one-versus-all" scheme, shown to be "extremely powerful and often at least as accurate as other methods" (Rifkin and Klautau 2004, p 102). We selected the following predictor features (variables) to define the sentences:

1. Raw sentiment scores according to the general lexicon. A list of positive and negative opinion words determines the score (Blair-Goldensohn et al. 2008; Hu and Liu 2004). For example, words such as *great*, *fantastic*, and *thrilled* are positive opinion words, and words such as *damaged*, *flawed*, and *weak* are negative opinion words. A dictionary of around 6800 words, http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar, informed our empirical study. If a sentence has *n* positive opinion words and *m* negative words, its raw sentiment score is $n - m$.
2. Magnitude of opinion. This measure is calculated as an indicator of how strong the opinion is: $\text{Magnitude} = \frac{n-m}{n+m}$. When the magnitude measure is large, the opinion of the sentence is highly polarized.
3. User rating, or the star rating of the product provided by the reviewer. If the star rating is low, each sentence in the review is more likely to express a negative opinion.
4. Domain-specific textual features. Textual features were selected in two steps: The top 200 words with substantive meaning were selected using an *importance index* (Eliashberg et al. 2007). Words that appear in almost every review and words that appear in very few reviews should be screened out by the importance index. The index in our study is calculated using $I_i = \left(1 - \frac{\sqrt{d_i}}{D}\right)\sqrt{N_i}$. Here, d_i is the number of reviews containing the *i*th word, *D* is total number of reviews, and N_i is the total frequency of this word.

Any of these predictor features can be omitted if not available, but including all of them offers the best classification accuracy and thus the best indication of the sentiment. We compared four supervised machine learning algorithms (methods) (for more details, see Feldman and Sanger 2007; Hastie et al. 2009). We use S to denote the sentiment of a sentence, d_t to indicate a sentence in the training set with tagged sentiment label, d_i for a sentence with an unknown sentiment label, and $\boldsymbol{x_i} = (x_1, \ldots x_n)$ as a feature vector for sentence i.

The maximum entropy (ME) classifier determines a posterior probability distribution of the classes through linear functions in the input features. The ME principle is to choose a model consistent with all facts but otherwise as uniform as possible (Berger et al. 1996). For a binary response variable, the ME classifier is commonly known as a logistic regression classifier, which is a generalized linear model. The model specifies a log-odds (logit) transformation of the response with the form (Hastie et al. 2009):

$$\log \frac{\Pr(S=1|d_i)}{\Pr(S=0|d_i)} = \beta_0 + \boldsymbol{\beta}_1^T \boldsymbol{x_i}.$$

In our study, to detect the positive sentiment, an event $S = 1$ indicates that a sentence contains a positive sentiment, and $S = 0$ suggests that the sentence contains no positive sentiment. Each entry in $\boldsymbol{\beta}$ can be interpreted as the weight for the corresponding feature. The best model parameter vector $\boldsymbol{\beta}$ can be fitted using optimization algorithms such as iterative scaling, conjugate gradient, or the BFGS algorithm (Malouf 2002).

A classification or decision tree classifier builds a tree-structured flowchart to select the response class for a given set of input values. Starting from the root node, at each node of the tree, some condition checks a logical condition of one input feature and selects a child branch. On reaching a leaf or terminal node, a classification label gets assigned. For the construction of the decision tree, a feature gets chosen at each step, and the splitting decision ensures maximal information gain (Bird et al. 2009).

A naïve Bayes (NB) classifier is simple and based on the assumption that each feature is conditionally independent from all others. Although this assumption is obviously not true, the NB classifier functions well for text classification. According to the Bayes rule, the probability that a sentence belongs to a class is:

$$\Pr(S = s|d_i) = \frac{\Pr(d_i|S=s)\Pr(S=s)}{\Pr(d_i)}.$$

If the independence assumption holds, the most probable category for a sentence d_t can be calculated as:

$$S^* = \text{argmax}_{s \in S} \Pr(S = s) \prod_{j=1}^{n} \Pr(x_j | s).$$

Training a NB classifier requires estimating each probability. Counting the number of positive sentences in the training set, divided by the total number of sentences in the training set, yields the maximum likelihood estimate of $\Pr(S = postive)$. The probability of a feature, given the class label $\Pr(x_j | s)$, can be estimated from the training set using Laplace smoothing:

$$\Pr(x_j | s) = \frac{1 + \sum_{d_t \in s} \mathrm{I}(x_j, d_t)}{n + \sum_{j=1}^{n} \sum_{d_t \in s} \mathrm{I}(x_j, d_t)},$$

where $\mathrm{I}(x_j, d_t)$ equals 1 if feature x_j occurs in sentence d_t, and 0 otherwise. These equations assume that the features are binary, such as whether each word occurs in a sentence. Non-binary features, such as user ratings, can be converted into binary features by binning or replacing $\Pr(x_j | s)$ with the estimated normal density for each class (John and Langley 1995).

Support vector machines (SVM) originally were proposed by Cortes and Vapnik (1995); they have become very popular as a classification method, because of their scalability and performance. In the effort to find a separating hyper-plane between two classes, a nice property of SVM is that learning ability is independent of the dimensionality of the feature space (Joachims 1998). Therefore, this method is suitable for text classification, considering the high number of features usually contained in a text document (for details, see Burges 1998; for SVM's application in marketing, see Cui and Curry 2005).

To increase prediction accuracy, we created a bagging predictor for each method. Bagging is an ensemble learning method that combines multiple classifiers for prediction (Breiman 1996). In our empirical study, the training examples were bootstrapped over 15 rounds, and in each round, a separate classification model fit on the bootstrapped sample. A combined

classifier then formed by predicting new observations with the majority vote of the 15 classifiers. Bagging can produce more stable prediction results (reducing variance) and offer substantial gains in prediction accuracy (Breiman 1996; Hastie et al. 2009). The bagging procedure uses the following pseudo-code:

1. Given labeled sentences (d_1,S_1), ..., (d_m,S_m) in training set,
2. For t = 1, ..., T = *15*:
   i. Bootstrap training set by selecting m random examples from the training set with replacement.
   ii. Train classifier h_t using the bootstrapped sample.
3. The bagging classifier is: $H(d_i)$ = majority($h_1(d_i)$, ..., $h_T(d_i)$).

In rare cases (.34% of observations), positive and negative predictors give conflicting predictions. We used a voting method; the number of individual classifiers inside the bagging procedure with positive versus negative predictions served as the tie breaker. For example, if 10 of the 15 positive classifiers predict positive and 8 of the 15 negative classifiers predict negative, the sentence would be classified as positive.

To train the classification tree and NB classifiers, we used the Python Natural Language Toolkit (Bird 2006), whereas for the SVM and ME classifiers, we used Scikit-learn (Pedregosa et al. 2011). Both are freely available under open source licenses. The accuracy measures were estimated using 10-fold cross-validation. Table 2 presents the performance of the learning algorithms for sentiment classification. We chose the SVM in our empirical study, noting its overall accuracy. The SVM classifier achieves precision of .741 and recall of .685 for positive sentiments, as well as precision of .727 and recall of .615 for negative sentiments. Of all the sentences that the classifier determines to be positive, 74.1% also were identified as positive by raters; of all the sentences that expressed positive sentiments toward an attribute, the classifier detected 68.5% of them. We considered the sentiment classifier's accuracy fairly high, because sentiments expressed in free-form reviews are sometimes ambiguous even for human raters. The

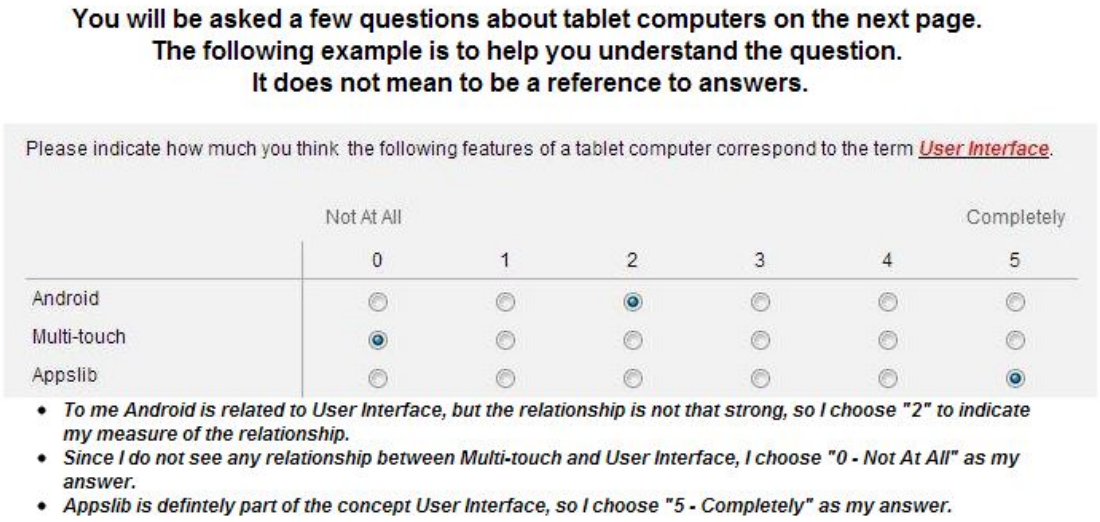
accuracy measures in our study are comparable to or better than those for sentence-level sentiment analyses in extant research (Gamon et al. 2005; Meena and Prabhakar 2007; Täckström and McDonald 2011).

Table 2. Performance of Sentiment Classifiers

	Positive				Negative			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Maximum entropy	.748	.716	.682	.699	.775	.709	.607	.654
Classification tree	.725	.728	.625	.673	.755	.691	.545	.609
Naïve Bayes	.738	.712	.675	.693	.744	.695	.587	.636
SVM	.766	.741	.685	.712	.781	.727	.615	.666

Appendix B. Web-Based Survey to Evaluate Meta-Attributes

We recruited 179 undergraduate students from a Midwestern U.S. university, who completed a web-based survey for extra course credit. Each participant considered three, randomly selected meta-attributes, each of which contained 10 individual product attributes that had been randomly generated from the list of 93 attributes. Participants evaluated how much the individual product attributes corresponded to the provided meta-attributes on a five-point scale, where 0 represents no correspondence at all and 5 indicates full correspondence. The survey also included instructions to help students to understand the question, as in the following screen shot.



References

Agrawal, R. and R. Srikant (1994), "Fast algorithms for mining association rules," *Proceedings of the 20th International Conference on Very Large Data Bases*, 487-99.

Archak, N., A. Ghose, and P.G. Ipeirotis (2011), "Deriving the pricing power of product features by mining consumer reviews," *Management Science*, 57 (8), 1485-509.

Berger, A.L., V.J.D. Pietra, and S.A.D. Pietra (1996), "A maximum entropy approach to natural language processing," *Computational linguistics*, 22 (1), 39-71.

Bird, S. (2006), "NLTK: the natural language toolkit," *Proceedings of the COLING/ACL on Interactive presentation sessions*, 69-72.

Bird, S., E. Klein, and E. Loper (2009), *Natural language processing with Python*: O'reilly.

Blair-Goldensohn, S., K. Hannan, R. McDonald, T. Neylon, G.A. Reis, and J. Reynar (2008), "Building a sentiment summarizer for local service reviews," *WWW Workshop on NLP Challenges in the Information Explosion Era (NLPIX)*.

Breiman, L. (1996), "Bagging predictors," *Machine learning*, 24 (2), 123-40.

Burges, C.J. (1998), "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, 2 (2), 121-67.

Cortes, C. and V. Vapnik (1995), "Support vector machine," *Machine learning*, 20 (3), 273-97.

Cui, D. and D. Curry (2005), "Prediction in marketing using the support vector machine," *Marketing Science*, 24 (4), 595-615.

De Marneffe, M.-C. and C.D. Manning (2008a), "The Stanford typed dependencies representation," *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, 1-8.

De Marneffe, M.C. and C.D. Manning (2008b), "Stanford typed dependencies manual," *URL http://nlp. stanford. edu/software/dependencies_manual. pdf*.

Dunn, J.C. (1974), "Well-separated clusters and optimal fuzzy partitions," *Journal of cybernetics*, 4 (1), 95-104.

Eliashberg, J., S.K. Hui, and Z.J. Zhang (2007), "From story line to box office: A new approach for green-lighting movie scripts," *Management Science*, 53 (6), 881-93.

Feldman, R. and J. Sanger (2007), *The text mining handbook: advanced approaches in analyzing unstructured data*: Cambridge University Press.

Gamon, M., A. Aue, S. Corston-Oliver, and E. Ringger (2005), "Pulse: Mining customer opinions from free text," in *Advances in Intelligent Data Analysis VI*: Springer.

Geffet, M. and I. Dagan (2004), "Feature vector quality and distributional similarity," *Proceedings of the 20th international conference on Computational Linguistics*, 247-53.

Hastie, T., R. Tibshirani, and J. Friedman (2009), "The Elements of Statistical Learning: Data Mining, Inference, and Prediction, (Springer Series in Statistics)," Springer. New York, NY: Springer.

Hedley, J. (2010), "jsoup: Java html parser," [available at http://jsoup.org/].

Hu, M. and B. Liu (2004), "Mining and summarizing customer reviews," *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 168-77.

Joachims, T. (1998), "Text categorization with support vector machines: Learning with many relevant features," in *Lecture Notes in Computer Science*: Springer.

John, G.H. and P. Langley (1995), "Estimating continuous distributions in Bayesian classifiers," *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, 338-45.

Klein, D. and C.D. Manning (2003), "Accurate unlexicalized parsing," *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, 423-30.

Lin, D. (1998a), "Automatic retrieval and clustering of similar words," *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, 768-74.

---- (1998b), "An information-theoretic definition of similarity," *Proceedings of the 15th international conference on Machine Learning*, July, 296-304.

Malouf, R. (2002), "A comparison of algorithms for maximum entropy parameter estimation," *Proceedings of the Sixth Conference on Natural Language Learning*, 20, 49-55.

Marcus, M.P., M.A. Marcinkiewicz, and B. Santorini (1993), "Building a large annotated corpus of English: The Penn Treebank," *Computational Linguistics*, 19 (2), 313-30.

Meena, A. and T.V. Prabhakar (2007), "Sentence Level Sentiment Analysis in the Presence of Conjuncts Using Linguistic Analysis," in *Advances in Information Retrieval*, Giambattista Amati and Claudio Carpineto and Giovanni Romano, eds. Vol. 4425: Springer Berlin Heidelberg.

Pantel, P. and D. Lin (2002), "Discovering word senses from text," *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 613-19.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg (2011), "Scikit-learn: Machine learning in Python," *The Journal of Machine Learning Research*, 12 (Oct), 2825-30.

Punj, G. and D.W. Stewart (1983), "Cluster analysis in marketing research: Review and suggestions for application," *Journal of Marketing Research*, 20 (2), 134-48.

Rifkin, R. and A. Klautau (2004), "In Defense of One-Vs-All Classification," *The Journal of Machine Learning Research*, 5, 101-41.

Rousseeuw, P.J. (1987), "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, 20 (Nov), 53-65.

Subramaniam, L.V., S. Roy, T.A. Faruquie, and S. Negi (2009), "A survey of types of text noise and techniques to handle noisy text," *AND '09: Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data*, 115-22.

Täckström, O. and R. McDonald (2011), "Semi-supervised latent variable models for sentence-level sentiment analysis," *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, 569-74.

Turney, P.D. (2001), "Mining the web for synonyms: PMI-IR versus LSA on TOEFL," *Proceedings of the 12th European Conference on Machine Learning*, 491-502.

Turney, P.D. and M.L. Littman (2003), "Measuring praise and criticism: Inference of semantic orientation from association," *ACM Transactions on Information Systems (TOIS)*, 21 (4), 315-46.

Turney, P.D. and P. Pantel (2010), "From frequency to meaning: Vector space models of semantics," *Journal of Artificial Intelligence Research*, 37 (1), 141-88.

Wei, C.-P., Y.-M. Chen, C.-S. Yang, and C.C. Yang (2009), "Understanding what concerns consumers: a semantic approach to product feature extraction from consumer reviews," *Information Systems and e-Business Management*, 8 (2), 149-67.

WordPress.org "Codex:List of common misspellings," [available at http://codex.wordpress.org/Codex:List_of_common_misspellings].

Yi, J., T. Nasukawa, R. Bunescu, and W. Niblack (2003), "Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques," *Proceedings of the Third IEEE International Conference on Data Mining*, 427-34.

2 Dynamics of Market Segmentation via Deep Learning and Evolutionary Clustering

2.1 INTRODUCTION

In this essay, we propose two new models to study the dynamics of market segmentation. The goal of market segmentation research is to identify group of entities (consumers, markets, companies) that share certain characteristics in order to better understand their behavior (Maggetti et al. 2012). Allenby et al. (2002) point out that the ideal outcomes of the market segmentation are not only just segments, but "part of corporate culture, providing discrete labels for groupings, which organize managerial thinking and facilitate communication by providing concrete characterizations of consumer wants within a market." Through deep learning algorithms and evolutionary clustering, our framework can derive insights on the constant changing landscape of competitive market from user-generated content (UGC), namely customer reviews.

All segmentation research, regardless of the method used, is designed to identify groups of entities (people, markets, organizations) that share certain common characteristic (Punj and Stewart 1983). In Essay 1, we demonstrated the validity of a brand segmentation approach that

involves mining a large collection of customer reviews: we start by discovering salient product attributes, then build a hierarchical structure of product attributes, and measure consumer sentiments towards the more-abstract 'meta-attributes'. Lastly, the common characteristics of brands would be the similar consumer sentiment scores mapped to a 2-D space after multi-dimensional scaling.

The proposed method of market segmentation carries on the same underlying consumer-centric thought. We would like to study the cluster of brands, products, or customers directly from feedback left by customers. In marketing literature, cluster analysis has been one of the primary tools for market segmentation, and by tracing and understanding of the changes of the clusters of brands or customers, we can potentially study the following questions: Is a brand segment simply disappearing or are its members migrating to other segments? Is a new emerging segment of customers reflecting new group of users or does it rather consist of existing customers whose preferences and tastes shift?

Beyond these questions, as Plummer (1974) pointed out, market segmentation has been employed in the development of potential new product opportunities. By dynamically clustering brands or products, competitive sets within the larger market structure can be studied in real time. The firm can thus dynamically determine the extent to which a current or new product offering is uniquely positioned or is in a competitive set with other emerging product.

The basic idea of using UGC to study the dynamic segmentation is as follows. Since online customer reviews are often dated, the collection of text (corpus) we can collect contains a time dimension. Therefore, other than treating the corpus as a single and static sample, we can treat the reviews as a stream of textual data that are continuously generated as time passes by. An analysis of changes in the reviews over time can be achieved by selecting a time window size

and mapping intelligence gathered from all reviews selected within that time window. Then, based upon a chosen step size, we can slide the time window forward by one step and repeat the analysis. This is one of the various forms of data stream mining (Gaber et al. 2005). The results allow us to conduct subsequent analyses. For example, we can identify "significant" changes in brand segmentation over time. These changes can then be correlated with internal adjustments of cooperate strategies or new product introductions and other external shocks.

We need to resolve several key challenges when using data stream analytics for textual reviews. The first issue is data sparsitiy. If we wish to conduct a somewhat fine-grained analysis, in a given time window the number of reviews that *explicitly* mention the product attributes is very limited. However, more reviews would mention the attributes *implicitly*. For example, instead of mentioning the attribute 'wifi', a reviewer may write 'I keep having trouble connecting the device to my router'. Therefore, we need methods to match the sentences to the attributes according to the latent semantic connection between words such as wifi, router, connecting. The hierarchical clustering using semantic similarity, as we proposed in Essay 1, would be computationally infeasible in this case due the sheer amount of words in the general vocabulary. Second, changes in brand's attribute should not occur instantaneously but rather evolve over time (Rutz and Sonnier 2011). Insights generated in previous period or rolling window should partially apply to next period. In addition, UGC is known to be noisy and sparse, and the snapshots approach may suffer from high sampling volatility. We need models that can take previous data into account in order to get more robust results.

We use two models to resolve these issues. First, we introduce the skip-gram model and semantic word vectors (distributed representations of textual content) for product attributes. Skip-gram model belongs to a machine learning paradigm referred to as "deep learning" that

involves using layers of artificial neural networks to learn representation of data (LeCun et al. 2015). Not only the model resolves the data sparsity issue by learning the representations of new and implicit attributes, it is also able to generate higher quality attribute hierarchy. Our second model is the evolutionary clustering. Compared to classical market segmentation method, evolutionary clustering model allows us to conduct segmentation analysis at consecutive time points and thus allow the evolution of product or customer segments to be monitored. The estimation of evolutionary clustering is simpler than Rutz and Sonnier (2011)'s DLM model, which is based on brand choice data. It does not require future data for backward sampling. Therefore, it is well-suited for monitoring large volume of UGC in real time.

2.2 METHOD

2.2.1 Product Attribute Embedding Model

Capturing semantic information from text remains one of the greatest challenges in learning from natural language. Essay 1 adapts a classic method of measuring semantic similarity proposed by Lin (1998). The algorithm uses semantic vectors to quantitatively summarize all grammatical relationships concerning a focal phrase in text; in our case, a noun phrase describing a product attribute. Each entry in the semantic vector contains the association measure (e.g. pointwise mutual information or PMI) of a grammatical dependency relationship between an attribute and a related word. For example, using a tablet PC review corpus, the top three entries in a semantic vector for the phrase *USB cable* are: *charge*-dobj-3.50, *recharge*-dobj-3.41, and *connect*-dobj-3.34. These entries indicate that *USB cable* often serves as the direct object for the verbs *charge*, *recharge*, and *connect* in the review text. The PMI scores at the end of each entry, such as 3.50 for *charge*, indicate the relative importance of the relationship. The semantic vector

therefore suggests that the product attribute *USB cable* provides functions such as charging and connecting to other devices. Finally, the similarity between a pair of attributes can be calculated using a similarity measure, such as the cosine distance between vectors.

The drawbacks with a pure PMI approach are two-fold. The most obvious is the curse of dimensionality induced by synonyms, i.e. each attribute's usage situation can be expressed in scores of ways by reviewers. The dimensions can be abundant, resulting in sparse and low-quality semantic vectors. For example, one may argue that *charge* and *recharge* are redundant. Since the dimension of semantic vectors is the union of all contextual words for all attributes, the overall dimension of these vectors can be in the millions. Second, because dependency parsing is computationally intense, Lin's method is unduly time-consuming when applied to a large collection of reviews, especially for modern products with many high-dimensional attributes.

Our solution is to use the state-of-the-art neural embedding model introduced by Google researchers Mikolov et al. (2013a). We reduce high-dimensional representations of product attributes from reviews to a markedly lower-dimensional space that preserves as much of the properties of the original data as possible. The method is predictive, rather than count-based, and is extremely efficient at learning high-quality representations of product attributes from unstructured text; a property demonstrated by our empirical examples.

Specifically, we adopt the skip-gram model (Mikolov et al. 2013b) to represent semantic vectors. Intuitively, this model uses artificial neural networks to predict the context that surrounds a given product attribute. In our running example, given the noun-phrase *USB cable*, the trained model would predict words such as *charge*, *recharge*, and *connect* as its contexts. Formally, we wish to represent a product attribute w using a d dimensional semantic vector v_w. The skip-gram model seeks to maximize the log probability:

$$\frac{1}{|V|}\sum_{t=1}^{|V|}\sum_{-k\le j\le k, j\neq 0}\log p(w_{t+j}\,|w_t), \tag{1}$$

where k is the "window size" of the context (usually between 5 and 10 words), w_t is a phrase at location t (product attributes are nouns or noun phrases that comprise a subset of all phrases), $|V|$ is the size of the entire vocabulary V. If we let C be the set of all available contexts and d be the desired dimensionality of the semantic vector, then the goal is to choose parameters θ to maximize the corpus probability as shown in (2).

$$\begin{aligned} &\arg\max_{\theta}\prod_{w\in V}\prod_{c\in c(w)} p(c|w;\theta) \\ =\ &\arg\max_{\theta}\sum_{\text{All }(w,c)}\log p(c|w;\theta). \end{aligned} \tag{2}$$

In (2), $c \in c(w)$ is the set of all contexts for phrase w; θ consists of v_ws in $\mathbb{R}^d$ for $w \in |V|$, and v_cs in $\mathbb{R}^d$ for $c \in c(w)$, with a total of $|C| \times |V| \times d$ individual parameters.

In order to estimate the parameter vector θ, consider that a single-hidden-layer neural network first projects a phrase w to a vector v_w in $\mathbb{R}^d$. Then a multinomial logit model is trained using the v_w as the independent variables to predict the probability (3).

$$p(c|w;\theta) = \frac{\exp(v_c^{\mathrm{T}} v_w)}{\sum_{c'\in C}\exp(v_{c'}^{\mathrm{T}} v_w)}. \tag{3}$$

Here the $v_c s$ can be viewed as the βs in the usual multinomial logit model. Lastly, the log-likelihood of the entire model is computed by summing over all (w, c) combinations, resulting in equation (2). The learning of semantic vectors v_ws is achieved when the log-likelihood is maximized.

Levy and Goldberg (2014) show that a skip-gram model can be viewed as an implicit estimation of the best rank-*d* singular-value decomposition (SVD) approximation – shifted by a global constant – to the original matrix R of *product attributes* by *context*. The entries in R are

exactly the PMI scores used in Lin's method. That is, suppose R is an m by n matrix, it can be decomposed into

$$R = U\Sigma V^T \tag{4}$$

where U is an $m \times d$ *product attribute* by *usage situation* orthogonal matrix, V is an $n \times d$ *linguistic context* by *usage situation* orthogonal matrix, and Σ is a $d \times d$ diagonal matrix of weights. The dimension d of the attribute embedding indexes the d most important latent contexts of the product attributes; akin to retaining the top *d* eigenvectors in a principle components analysis. In sum, our embedding model extracts the usage situation concepts behind contextual words, but not the words themselves.

A naïve estimation using iterative optimization techniques on the neural networks can be computationally impractical for large numbers of reviews. An efficient approximation algorithm for the skip-gram model, known as negative sampling, is provided by the open-source *word2vec* package published by Google. It allows high-quality model training without using any dense matrix multiplications. Mikolov et al. (2013a) demonstrated the time complexity advantage of the skip-gram-negative-sampling method over the LDA method, which can be slow on large datasets due to the Bayesian estimation involved. The negative sampling algorithm was first introduced by Gutmann and Hyvärinen (2012) as a parameter estimation method for unnormalized probabilistic models. The difficulty of solving the attribute embedding model using an iterative optimization procedure lies in the computing of $\nabla p(c|w;\theta)$ from Eq. (3) due to the size of all the contexts C. Note that $\log p(c|w;\theta)$ in Eq. (2) can be written as

$$\log p(c|w;\theta) = \log \frac{\exp(v_c^{\mathrm{T}} v_w)}{\sum_{c' \in C} \exp(v_{c'}^{\mathrm{T}} v_w)}$$

$$= \log \exp(v_c^{\mathrm{T}} v_w) - \log \sum_{c' \in C} \exp\left(v_{c'}^{\mathrm{T}} v_w\right) \quad (5)$$

Negative sampling replaces (5) using the expression

$$\log \frac{1}{1 + \exp(-v_c^{\mathrm{T}} v_w)} + \sum_{i=1}^{n} \log \frac{1}{1 + \exp(v_{c_i'}^{\mathrm{T}} v_w)} \quad (6)$$

where c_i's are n negative samples randomly generated from a "noise distribution". The idea is that if the model is trained correctly, it should be good at distinguishing correct (w, c) pairs (which we can observe from review data) from the randomly generated (w, c_i')s. In our empirical study we used $n = 5$ as recommended by Mikolov et al. (2013a). The C implementation of negative sampling is available via *word2vec* (https://code.google.com/p/word2vec/), and the Python version is available in *gensim* (Řehůřek and Sojka 2010).

With the semantic vectors learned using the product attribute embedding model, we can compute the pairwise similarity between two attributes using the cosine similarity measure defined as $\frac{v_{w1}^{\mathrm{T}} v_{w2}}{\|v_{w1}\| \|v_{w2}\|}$, where v_{w1} and v_{w2} are the semantic vectors of two product attributes. The similarity matrix then serves as the input for a hierarchical clustering procedure purposed to construct an attribute hierarchy like the one illustrated in Essay 1.

2.2.2 Evolutionary Cluster

In addition to the static visualization of brand position, we apply Chi et al's (2009) evolutionary clustering model (Chakrabarti et al. 2006; Chi et al. 2009) to utilize the timestamp on each product review to capture changing market structure. Specifically, we consider the problem of product differentiation over time. Product differentiation means that a given product

offering is perceived to have unique characteristics/benefits that differ from those of its competitors. Differentiation can be achieved via usage experience, word-of-mouth, promotion, or via actual product characteristics (Dickson and Ginter 1987). By using cluster-based brand segmentation, we study how product clusters evolve.

The goal of evolutionary clustering is to establish a cluster solution at time t that is faithful to consumer sentiment at time t and also to sentiment in the most recent periods. For each period, the objective function comprises two components: 1) snapshot cost (CS), used to measure the quality of a solution in the current period; and 2) temporal cost (CT), which indexes deviations in sentiment from previous periods. The objective can be written as a linear combination of CS and CT:

$$Cost = \alpha CS + (1 - \alpha)CT. \tag{7}$$

where $0 \leq \alpha \leq 1$ is a user-defined smoothing parameter that controls the trade-off between the two costs. For a *k*-means clustering problem, the cost can be written as:

$$Cost = \alpha \sum_{l=1}^{k} \sum_{i \in C_{l,t}} \left\| x_{i,t} - \mu_{l,t} \right\|^2 + (1 - \alpha) \sum_{l=1}^{k} \sum_{i \in C_{l,t}} \left\| x_{i,t-1} - \mu_{l,t-1} \right\|^2, \tag{8}$$

where $x_{i,t}$ is the attribute sentiment vector for brand i at time t, and $C_{l,t}$ is the set of brands in cluster l at time t. For the CT part of the cost function, the inner summation is based on the cluster partition at time t, but the sentiment vectors and cluster means used are from recent solutions. Therefore, the cost function penalizes deviations in the composition of current segments from those of segments in the recent past. The optimization problem using the cost function (8) is NP-hard. We resort to spectral clustering (Zha et al. 2001) to solve a relaxed version of the problem and obtain approximate solutions. Specifically, the solution that minimizes cost in (8) can be approximated using spectral clustering. Suppose S is a n by m

sentiment matrix of n brands and m attributes. We first compute the inner product of the brand by sentiment matrix $= SS^T$. W can be considered as a similarity matrix between brands given the usual k-means objective. Then we compute the top k eigenvectors of $\alpha W_t + (1-\alpha)W_{t-1}$ and let them be the columns of an n by k matrix X. With X, sentiment data is projected onto the spectral domain where brands can be more separable. Lastly, running a k means clustering on X gives an approximate solution to the original (8).

A challenge posed by analyzing market structure over time is that new brands enter and other brands exit during the transition from one analysis period to another. To handle this issue, we apply heuristics suggested by Chi (2009). For brands in $t-1$ that are no longer of significance in period t, the corresponding rows and columns of W_{t-1} are excluded from the calculation. For brands that appear in period t but were not present in period $t-1$, we impute similarity scores for period *t* as follows. Denote n_2 as the number brands in period t, n_1 as the number of brands in both periods, then the corresponding $\widehat{W_{t-1}}$ can be extended as:

$$\widehat{W_{t-1}} = \begin{bmatrix} W_{t-1} & E_{t-1} \\ E_{t-1}^T & F_{t-1} \end{bmatrix} \quad (9)$$

where W_{t-1} is the brand by brand similarity matrix of common brands in both periods, and $E_{t-1} = \frac{1}{n_1} W_{t-1} \mathbf{1}_{n_1} \mathbf{1}_{n_2-n_1}^T$, $F_{t-1} = \frac{1}{n_1^2} \mathbf{1}_{n_2-n_1} \mathbf{1}_{n_1}^T W_{t-1} \mathbf{1}_{n_1} \mathbf{1}_{n_2-n_1}^T$. These calculations assume that brands in period *t*-1 have average similarity with brands included in period *t*.

2.3 DATA

The dataset is collected from Amazon.com using a Java crawler published by Wang et al. (2014) in March 2014 . Reviews were cleaned and then preprocessed to tag parts-of-speech and to parse syntactic dependencies. The resulting analysis dataset includes 306 brands and 1,503

tablet devices (distinct SKUs). Reviews contained an average of 9.14 sentences. Review sentences consisted of 15.4 words on average. In total, we analyzed 736,224 review sentences and 11,337,851 words. Table 1 and Table 2 describe the variables and structure of our dataset.

Table 1 Variables of Products and Brands

Product	Item_ID	Amazon Standard Identification Number (ASIN): Amazon assigns a unique identification number to each product
	Title	Title of the product
	Brand	Brand name of the product
	Model	Model number provided by the manufacturer
	UPC	Universal product code of the product

Table 2 Variables of Consumer-Generated Product Reviews

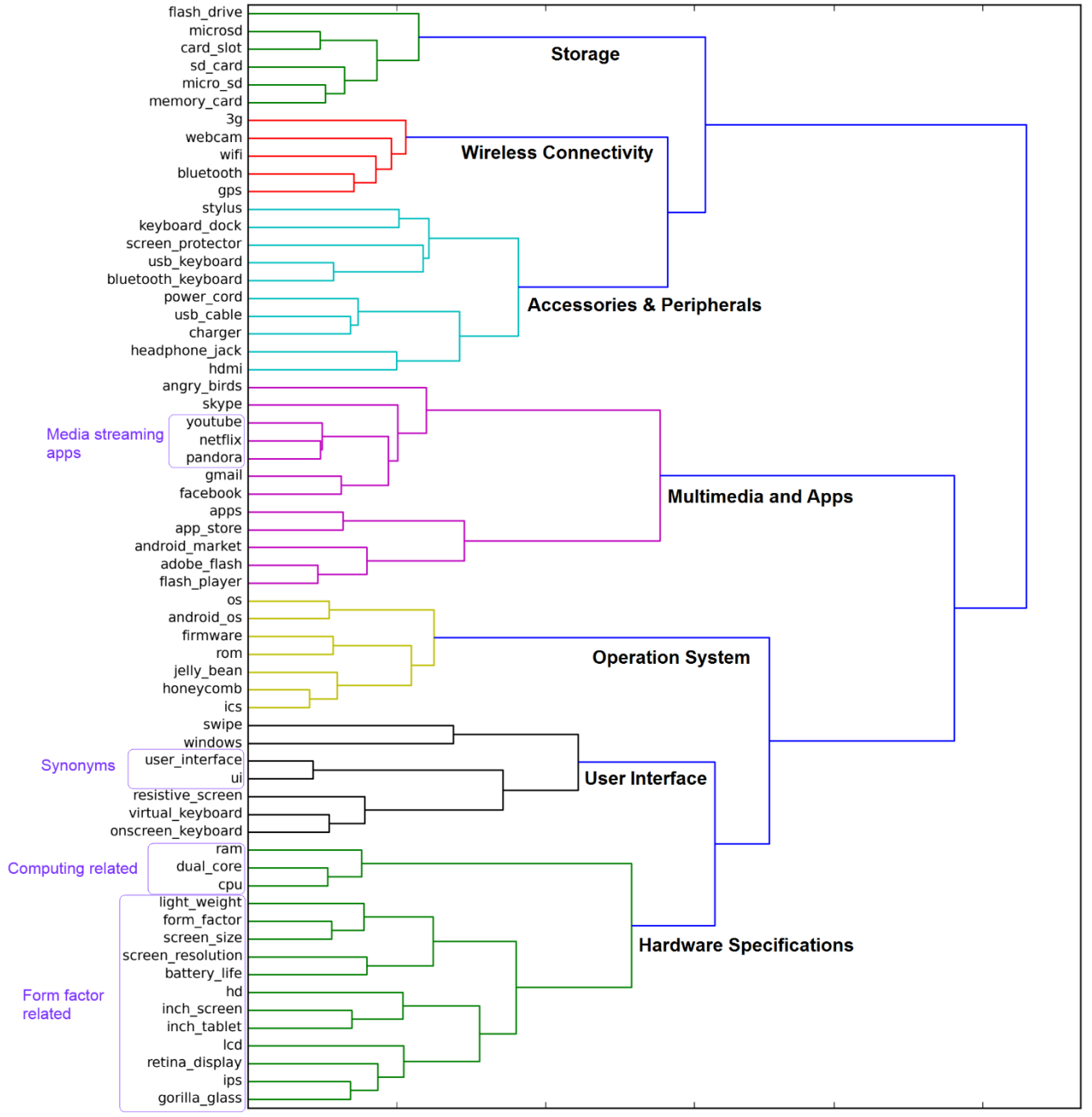
Variable Name	**Description**
Review_ID	Unique identification number for each review
Item_ID	Amazon Standard Identification Number (ASIN) of the product
Reviewer_ID	Unique reviewer identification number assigned by Amazon
Review_date	Date the review was submitted
Title	Title of the review
Review	Textual content of the review
Rating	Numerical rating of the review (1–5 stars)

2.4 RESULTS

2.4.1 Attribute Hierarchy and Implications

A filtering procedure similar to what was described in Essay 1 was used to extract 62 product attributes. Next, the product attribute embedding algorithm was implemented using *gensim* (Řehůřek and Sojka 2010) with dimension size $d = 100$, and window size $k = 5$. A similarity matrix was then constructed using the cosine distance between embedding vectors. The dendrogram (Figure 1) was generated using an agglomerative clustering procedure on this similarity matrix. For this study, we chose seven meta-attributes (i.e., seven clusters) following the reasoning described in Essay 1.

Figure 1 Dendrogram of Tablet's Attribute Hierarchy Constructed via Embedding Model



Besides the seven-cluster solution, we could also cut the dendrogram of product attributes at a higher level, which would result in fewer meta-attributes, or at a lower level, which would provide more meta-attributes. For example, consider the Hardware Specifications meta-attributes. A meaningful hierarchical structure exists within it. We can differentiate the three attributes at the top – RAM, Dual Core, CPU, corresponding to the function that concerns the speed of computing. While the larger group of attributes at the bottom, which correspond to the form factor and screen types.

A helpful aspect of the neural embedding model is that it computes the vector representation of all phrases in the reviews. Therefore, given a product attribute that is not explicitly mentioned by consumer, as long as we have a meaningful product attribute hierarchy, we can simply compute its similarity to the closest bundle and place it accordingly. This property effectively solves the data sparsity issue we described in the introduction. It is also a useful tool for marketing managers to keep track of a fast changing product market. For example, the *AMOLED* (active-matrix organic light-emitting diode) is a recent advancement in display technologies. Without including it in the original attribute hierarchy (Figure 1), the neural embedding model identified the closest attribute to *AMOLED*. The result is *LCD* (liquid crystal display), with a cosine similarity index of 0.857. (In Figure 1, note that LCD is a first-order dimension in the Hardware Specs meta-attribute along with display properties, form and weight factors.) This ability means that marketers can use our model to recursively (vs. batch) update an attribute hierarchy model as new reviews become available.

Other than constructing the attribute hierarchy, the attribute neural embedding model is able to illuminates the needs of consumers via vector algebra using the product attributes as information carriers. To demonstrate this, consider one lower level meta-attributes under

Multimedia & Apps which consists of YouTube, Netflix, and Pandora in Figure 1. YouTube and Netflix are video platforms used to satisfy the need of watching video, and Pandora focuses on music and audio streaming. The five attributes that are closest to Pandora are listed in Panel a, all related to audio streaming. As a second example, we can compute the average semantic vector from the three individual attributes and use the mean vector as a representation of the overall needs of media streaming. The five attributes that are most similar to this mean vector is listed in Panel b. These attributes are a mixture of both audio and video content providers. More interestingly, we can compute the vector difference between the video providers and the audio providers by taking the average of YouTube and Netflix vectors, and subtracting Pandora vector. Panel c lists the product attributes that are most similar to the resulting difference vector. All of the attributes in Panel c are heavily video focused.

Table 3 The most similar attributes after certain vector computations

Attributes	**Semantic Similarity with "Pandora"**
iHeartRadio	0.841
Audiobooks	0.837
Spotify	0.837
Podcast	0.835
ESPN (radio)	0.835

Panel a

Attributes	**Semantic Similarity with the Average of "Media Streaming Apps"**
HBO Go	0.841
Hulu	0.837
TV Show	0.837
Podcast	0.835
ESPN (radio)	0.835

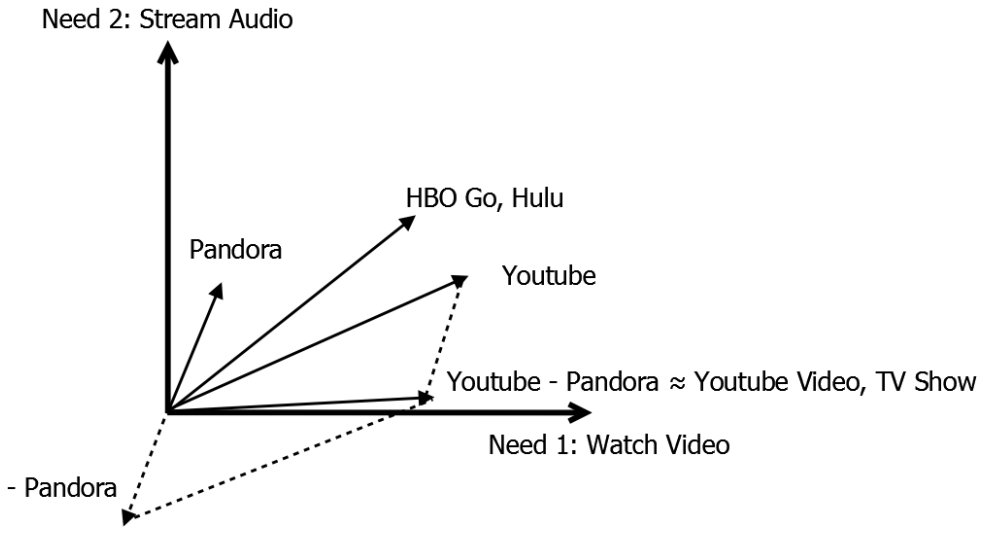
Panel b

Attributes	Semantic Similarity with [Youtube, Netflix] – [Pandora]
Video	0.738
YouTube Video	0.727
Hulu	0.708
Amazon Instant	0.705
TV Show	0.689
Video Stream	0.688
Vimeo	0.666

Panel c

Figure 2 provides an intuitive explanation of our computational results using a simplified 2d vector space (as opposed to the 100 dimension representation we used). The bases vectors correspond to "watch video" and "stream audio". The difference between a video-focused attribute, such as YouTube and an audio-focused attribute, such as Pandora, yields vectors that are most proximate to this contrast. In the case shown, these are Video, YouTube_video, Hulu, TV_show, Video_stream, and Vimeo. These solutions contrast with Pandora in the same way that YouTube does; each satisfies the need to "watch video" (vs. stream audio).

Figure 2 A simplified illustration of vector computation in the consumer needs space



Our method also recognizes attribute synonyms that consumers use to refer to the same product attribute. For example, *UI* and *user interface* are grouped in the same clusters at the first

level of aggregation. Consumers used these noun phrases interchangeably in their tablet reviews. Attribute synonyms and product usage situations help facilitate communication with target consumers, by using their own language.

Although the attribute embedding model does not explicitly generate the usage situations or needs that are associated with the meta-attributes. Table 4 presents the post-hoc analysis of common contexts of these seven meta-attributes. It shows that such a post-hoc analysis for our seven meta-attributes strongly supports the labels selected.

Table 4 Common Usage Situations of Seven Meta-Attributes

Meta-Attributes		Common Usage Situations
Storage	:	connect, plug, add, read, insert, purchase
Wireless Connectivity	:	turn, include, try, set, access, disconnect
Tablet Accessories & Peripherals	:	include, order, connect, put, carry, require
Multimedia and Apps	:	watch, play, download, load, view, browse
Operating System	:	update, upgrade, install, flash, release, root
User Interface	:	plug, display, recognize, support, customize, register
Hardware Specifications	:	love, enjoy, match, control, increase, compare

2.4.2 Validation of the Attribute Hierarchy

We assessed the quality of the attribute hierarchy with a web-based survey (see Cimiano and Staab (2005)). We asked 179 participants via Amazon's MTurk to evaluate the relationships among the seven meta-attributes and individual attributes. Only participants who had an Amazon MTurk approval rating of 95% or higher and lived in United States were permitted to participate. In the survey, participants considered random pairs of meta-attributes and attributes and rated the

level of correspondence on a five-point scale, where 5 represents the highest level of correspondence. In order to evaluate both discriminant and convergent validity of the meta-attributes, a product attribute may or may not correspond to a particular meta-attribute in the survey. For example, raters are equally likely to be asked to rate the level of correspondence between *MicroSD* and Storage, as between *MicroSD* and other six meta-attributes.

In Table 5, we summarize the percentage of ratings greater than or equal to 3 for each meta-attribute, which provided the similarity measure between lower-level attributes and a higher-level meta-attribute. We include both the similarity scores for attributes within the meta-attribute clusters generated by our method and scores for attributes beyond the meta-attribute clusters. The relatively high percentages show that the correspondence between meta-attributes and attributes is reasonable, according to the survey participants. The only meta-attribute that have convergent validity below 0.7 is Wireless connectivity, where most human raters rates "Webcam" and "GPS" not belonging to the category. In addition, the lower similarity measures in the right column suggest that our method effectively differentiates attributes unrelated to the abstraction represented by the meta-attributes. The measures in Table 5 have much lower "cross-loading" compared with the results in Essay 1.

Table 5 Empirical Evaluation Results of Seven Meta-Attributes

Meta-Attributes	**Similarity with Attributes in Cluster**	**Similarity with Attributes outside Cluster**
Storage	1.00	.036
Wireless Connectivity	.600	0
Tablet Accessories & Peripherals	.900	.154
Multimedia and Apps	.916	.04
Operation System	.714	036
User Interface	.857	.145
Hardware Specifications	.867	.091

We tested the effects of two parameters of the attribute embedding model as a robustness check; the context window size k and the dimensionality of the vector d. For each combination of k and d, we computed the correlation between two matrices: S_1, (the solution used in the main text) defined as the cosine similarity matrix between product attributes generated using $k = 5$ and $d = 100$; and S_2 the cosine similarity matrix generated using the given k and d. As shown in Table 6, when the context window size and dimension are sufficiently large, results are robust to the particular choice of k and d.

Table 6 Robustness under Different Context Window Sizes and Vector Dimensionalities

		Dimension			
		4	**40**	**200**	**500**
Window Size	**2**	0.264	0.852	0.916	0.915
	10	0.497	0.892	0.950	0.947

All p-values < 0.01.

2.4.3 Comparison with Latent Dirichlet Allocation

We now discuss and compare our results with Latent Dirichlet Allocation (LDA), a Bayesian topic modeling approach, which Tirunillai and Tellis (2014) use to extract quality dimensions from product reviews. An advantage of LDA is that it places few restrictions on the form, structure, or grammatical correctness of reviews because of the bag-of-words model. However, as with CA, LDA tracks the valence of dimensions (not the valence of specific nouns or noun phrases) and uses the heterogeneity of dimension valence to distinguish brands. In other words, LDA ignores consumer sentiment at the level of "localized" speech.

Topic modeling is used to automatically discover the index of ideas contained in the documents and identify which documents are about the same kinds of ideas (Blei and Lafferty 2009). The LDA (Blei et al. 2003) model assumes that the procedure of producing a review can be decomposed into a number of simple probabilistic steps. The statistical inference based on

hierarchical Bayesian analysis can then uncover the semantic structures in the texts and discover patterns of word usage. There are many advantages provided by the LDA model compared to more traditional text analytic models such as naïve Bayes classification and Latent Semantic Analysis (LSA). For example, LDA is built upon a rigorous foundation of Bayesian statistical inference and therefore has more principled model fitting and selection procedures. It provides "soft" classification for documents and therefore allows each document to be a multi-membership mixture of different topics. LDA also extends the ideas of probabilistic latent semantic analysis (PLSA) (Blei and Lafferty 2009) and can automatically learn contexts of word usage without recourse to a dictionary or thesaurus (Hofmann 2001).

There are several statistical assumptions inherent in the LDA model. The first major assumption of the LDA is the "bag of words" model, which means that the words appearing in the reviews are assumed to be exchangeable. Therefore, when applying statistical topic models such as LDA, we represent each review as a vector of word counts and neglect the order of the words. Given the "bag of words" assumption, the LDA model further assumes that: 1) words contained in each review are generated from a mixture of topics; 2) each topic has a probability distribution over a fixed word vocabulary; 3) the topics are shared by all of the review, but the topic proportions differ across review. Formally, LDA can be described using a generative process. It assumes that there are K different topics (the parameter K can be chosen using model selection techniques) and the vocabulary size is V. Each topic is associated with a Dirichlet distribution over all words in the vocabulary with parameters $\boldsymbol{\beta}$. For all topics $k \in 1 \ldots K$ the process first draws a vocabulary mixture $\boldsymbol{\phi}_k$ for the topic from Dirichlet ($\boldsymbol{\beta}$). Then, each review $m \in 1 \ldots M$ is assumed to be produced from the following generative process:

1. Sample length of the review N_m from a Poisson distribution with parameter ξ.

2. Sample topic proportions $\boldsymbol{\theta}_m$ from a Dirichlet distribution with parameters $\boldsymbol{\alpha}$.
3. For each of the $n \in 1 \ldots N$ words in m:
    a. Sample a topic assignment $z_{m,n}$ from Multinomial ($\boldsymbol{\theta}_m$), where $z_{m,n}$ is a topic index between $1 \ldots K$.
    b. Choose a word $w_{m,n}$ from Multinomial ($\boldsymbol{\phi}_{z_{m,n}}$).

The objectives of topic modeling can be viewed as reversing the above generative process using Bayesian inference (Blei 2012). We wish to infer the topic mixture of each review $\boldsymbol{\theta}_m$, and the word distributions of each topic $\boldsymbol{\phi}_k$. The former parameters indicate which topic(s) are covered in a given review, while the latter parameters tell us the representative words for each topic. Approximate inference algorithms such as Gibbs sampling (Steyvers and Griffiths 2006) and variational methods (Blei et al. 2003; Teh et al. 2006) have been developed, as exact inference is intractable for the model. Heinrich (2005) presents a detailed discussion of various parameter estimation methods for LDA.

We used the online learning algorithm outlined in Hoffman et al. (2010) to approximate the posterior distribution. Two hyperparameters α and β in the LDA model control the smoothing for document-topic distributions and topic-term distributions respectively. A smaller β generates more fine-grained topics and a smaller α tends to assign fewer topics to a document. We used symmetric prior $= \alpha = 1/K$. The optimal number of topics K was chosen to minimize perplexity, a widely-used performance metric that gives useful characterization of the predictive quality of a language model and correlates with other measures well (Asuncion et al. 2009).

An advantage of LDA is that it operates in an unsupervised manner, but the quality of results is low as shown in Table 7. The table shows results from the optimum six-dimensional solution discovered in our review corpus by the LDA model. As we see, LDA is able to extract important

latent topics from our reviews but because LDA cannot generate an attribute hierarchy, an analyst would not know whether these dimensions are first-order dimensions or higher-order meta-attributes. Results are also confusing because the phrases that LDA uses to best represent dimensions are not mutually exclusive. Thus, *awesome* is recovered as part of an awesome gift for dimension 2 and also for dimension 4 to describe an awesome web surfing experience. Other results are hard to integrate with the traditional market structure framework. For example, dimension 5 (aspects of customer service) refers to channel characteristics not brand characteristics. Finally, as Tirunillai and Tellis (2014) noted, LDA may not offer sufficient differentiation to conduct market structure analysis for vertically differentiated markets, such as the market for tablet computers or other electronic devices. We believe that, LDA is more suitable as a strategic analysis tool than as a way to identify an attribute ontology.

Table 7 Dimensions Extracted from Our Tablet Corpus Using LDA

Dimension/Topic		Representative Phrases
1	:	touch, touch screen, power, open, item
2	:	kid, wife, learn, card, free
3	:	movie, full, application, install, add
4	:	version, access, website, call, awesome
5	:	cheap, send back, customer service, replace, hour
6	:	light, OS, hand, amazing, hold, compare, performance

2.4.4 Evolutionary Clusters

To demonstrate evolutionary market structure analysis, we investigated the period from Jan 1, 2011 to Jan 1, 2014. We took the top 15 brands by number of reviews for the 13 quarters in this time span. A total of 38 brands qualified, but because the market was evolving dramatically during this time period, only 5 brands – Apple, ASUS, Coby, Lenovo, and Samsung – present in quarter one remained present in quarter 13. For our evolutionary clustering we concatenated quarters to six half-year periods and used α=0.8 for the smoothing parameter. Within a period, the optimal number of clusters was determined using the Bayesian information criterion (BIC) from model-based clustering (Fraley and Raftery 2002). The optimal number of recovered clusters varied from 2 to 4 across our time periods.

Table 8 Evolutionary Clustering of Top Tablet Brands

7/1/2011		1/1/2012		7/1/2012		1/1/2013		7/1/2013		1/1/2014	
Acer	1	Acer	1	Apple	1	Apple	1	Apple	1	Apple	1
Apple		Apple		Asus		Asus		Asus		Asus	
Asus		Asus		Samsung		Samsung		Coby		Dell	
Motorola	2	Samsung		Toshiba		Acer	2	Samsung		LeapFrog	
Samsung		Toshiba		Acer	2	Microsoft		Sony	2	Microsoft	
ViewSonic		BlackBerry	2	BlackBerry		AGPtek	3	Acer		Samsung	
BlackBerry	3	Coby		HP		Archos		Lenovo		Acer	2
Coby		HP		Archos	3	BlackBerry		Microsoft		Chromo	
TomTom		Lenovo	3	Coby		Coby		BlackBerry	3	Coby	
Velocity		Velocity		Lenovo		Fuhu		Chromo		DoublePower	
Archos	4	VTech		Motorola		LeapFrog		Digital2		Ematic	
Dell		Motorola	4	Polaroid		Lenovo		Fuhu		Fuhu	
HP		Sony		Sony		Sony		LeapFrog		Hannspree	
HTC		ViewSonic		Velocity		VTech		Matricom		Lenovo	
Lenovo		VIZIO		VIZIO		ZTO		Pipo		Matricom	

Figure 3 Characteristics of Clusters across Time

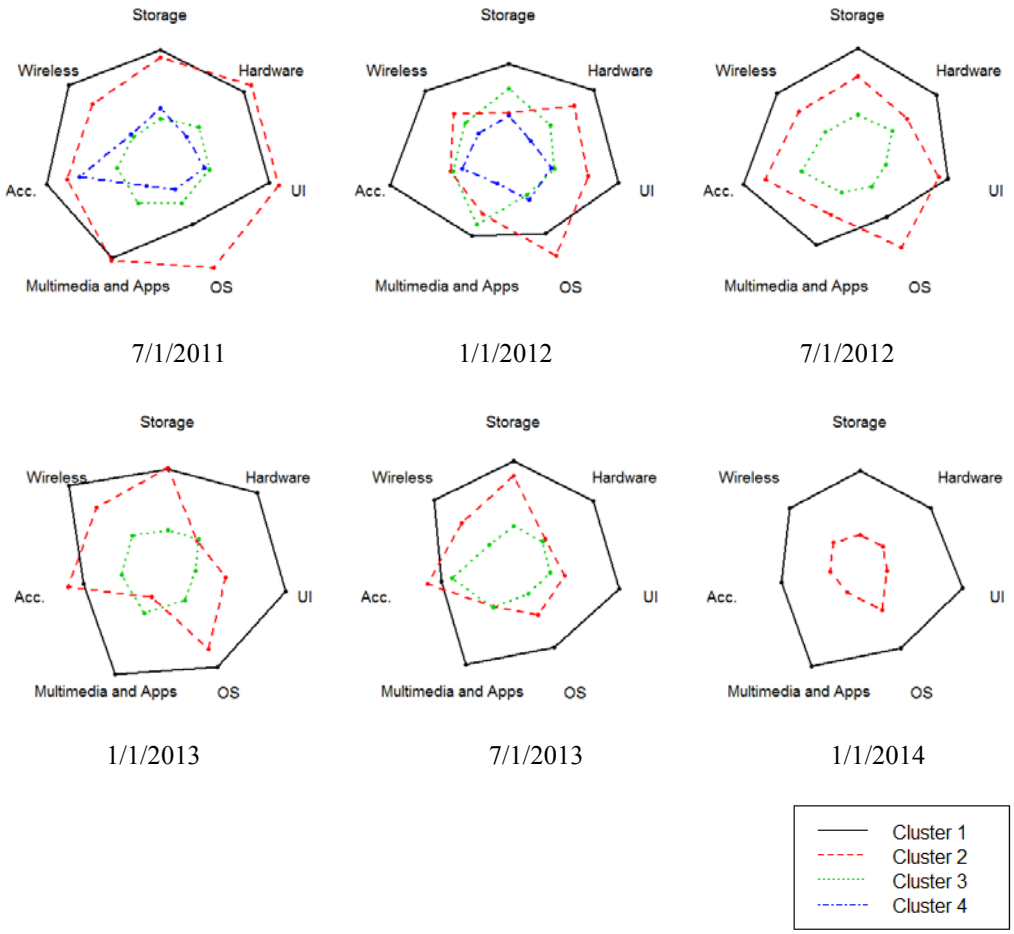


Table 8 shows cluster membership and each cluster's profile. Figure 3 contains cluster profiles in terms of the seven meta-attributes using radar charts. A profile is a cluster's centroid (average sentiment scores, by meta-attribute). For each period the clusters are arranged in descending orders according to the average sentiment ratings. Managers can trace through results periods-by-periods and drill down on the reasons of the market structure changes. For example, without external shocks, despite firm existing and entering, the characteristics of clusters across most periods remain similar. However, the shape of Cluster 1 (leading brands) noticeably changed after the introduction of the Windows 8 operating system in the second half of 2012.

Firms such as Dell made strategic choice to focus on Windows tablets and emerged from cluster 4 to cluster 1.

2.5 CONCLUSION AND FUTURE RESEARCH

In this essay we outline two new models to extract information from product reviews. The skip-gram model provides vectorized representation of key entities (in our case, product attributes) by learning consumer needs from reviews using deep learning. We also demonstrate how the needs of consumers can be computed via vector algebra using product attributes as information carriers. We compare our method with the LDA model, which also specializes in summarizing meanings in an unsupervised fashion from free-texts, and assess the limitations and strengths of our method. Our evolutionary clustering routine allows analyzing market structure using streams of product reviews and provides a potential a strategic tool for marketing managers

This essay investigates the application of one specific deep learning model, namely the skip-gram model for word embedding. Other recently-developed deep learning models can express variable-length pieces of texts, such as sentences, paragraphs, and documents using fixed-length vectors (Le and Mikolov 2014). These models can be used to capture information outside of the local attribute contexts of product reviews. In addition, the deep learning techniques combined with evolutionary clustering shall have applications beyond product reviews. It can be applied to other UGC such as microblogging, social media data, etc., in which time-dependent, domain specific knowledge can be discerned from multiword windows.

REFERENCES

Allenby, G., G. Fennell, A. Bemmaor, V. Bhargava, F. Christen, J. Dawley, P. Dickson, Y. Edwards, M. Garratt, and J. Ginter (2002), "Market segmentation research: Beyond within and across group differences," *Marketing Letters*, 13 (3), 233-43.

Asuncion, A., M. Welling, P. Smyth, and Y.W. Teh (2009), "On smoothing and inference for topic models," *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 27-34.

Blei, D.M. (2012), "Probabilistic topic models," *Communications of the ACM*, 55 (4), 77-84.

Blei, D.M. and J.D. Lafferty (2009), "Topic models," *Text mining: classification, clustering, and applications*, 10, 71.

Blei, D.M., A.Y. Ng, and M.I. Jordan (2003), "Latent dirichlet allocation," *the Journal of machine Learning research*, 3, 993-1022.

Chakrabarti, D., R. Kumar, and A. Tomkins (2006), "Evolutionary clustering," *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 554-60.

Chi, Y., X. Song, D. Zhou, K. Hino, and B.L. Tseng (2009), "On evolutionary spectral clustering," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3 (4), 17.

Cimiano, P. and S. Staab (2005), "Learning concept hierarchies from text with a guided hierarchical clustering algorithm," *Proceedings of Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods, ICML05*, 6-16.

Dickson, P.R. and J.L. Ginter (1987), "Market segmentation, product differentiation, and marketing strategy," *The Journal of Marketing*, 1-10.

Fraley, C. and A.E. Raftery (2002), "Model-based clustering, discriminant analysis, and density estimation," *Journal of the American Statistical Association*, 97 (458), 611-31.

Gaber, M.M., A. Zaslavsky, and S. Krishnaswamy (2005), "Mining data streams: a review," *ACM Sigmod Record*, 34 (2), 18-26.

Gutmann, M.U. and A. Hyvärinen (2012), "Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics," *The Journal of Machine Learning Research*, 13 (1), 307-61.

Heinrich, G. (2005), "Parameter estimation for text analysis."

Hoffman, M., F.R. Bach, and D.M. Blei (2010), "Online learning for latent dirichlet allocation," *advances in neural information processing systems*, 856-64.

Hofmann, T. (2001), "Unsupervised learning by probabilistic latent semantic analysis," *Machine learning*, 42 (1-2), 177-96.

Le, Q.V. and T. Mikolov (2014), "Distributed representations of sentences and documents," *arXiv preprint arXiv:1405.4053*.

Lecun, Y., Y. Bengio, and G. Hinton (2015), "Deep learning," *Nature*, 521 (7553), 436-44.

Levy, O. and Y. Goldberg (2014), "Neural word embedding as implicit matrix factorization," *Advances in Neural Information Processing Systems*, 2177-85.

Lin, D. (1998), "Automatic retrieval and clustering of similar words," *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, 768-74.

Maggetti, M., C.M. Radaelli, and F. Gilardi (2012), *Designing research in the social sciences*: Sage.

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013a), "Efficient estimation of word representations in vector space," in Proceedings of Workshop at ICLR.

Mikolov, T., I. Sutskever, K. Chen, G.S. Corrado, and J. Dean (2013b), "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, 3111-19.

Plummer, J.T. (1974), "The concept and application of life style segmentation," *the Journal of Marketing*, 33-37.

Punj, G. and D.W. Stewart (1983), "Cluster analysis in marketing research: Review and suggestions for application," *Journal of Marketing Research*, 20 (2), 134-48.

Řehůřek, R. and P. Sojka (2010), "Software framework for topic modelling with large corpora."

Rutz, O.J. and G.P. Sonnier (2011), "The evolution of internal market structure," *Marketing Science*, 30 (2), 274-89.

Steyvers, M. and T. Griffiths (2006), "Probabilistic topic models," in *Latent Semantic Analysis: A road to meaning*, T. Landauer and D Mcnamara and S. Dennis and W. Kintsch, eds.: Lawrence Erlbaum.

Teh, Y.W., D. Newman, and M. Welling (2006), "A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation," *Advances in neural information processing systems*, 1353-60.

Tirunillai, S. and G.J. Tellis (2014), "Mining Marketing Meaning from Online Chatter: Strategic Brand Analysis of Big Data Using Latent Dirichlet Allocation," *Journal of Marketing Research*, 51 (4), 463-79.

Wang, X.S., F. Mai, and R.H. Chiang (2014), "Database submission-market dynamics and user-generated content about tablet computers," *Marketing science*, 33 (3), 449-58.

Zha, H., X. He, C. Ding, M. Gu, and H.D. Simon (2001), "Spectral relaxation for k-means clustering," *Advances in neural information processing systems*, 1057-64.

3 Model-Based Capacitated Clustering with Posterior Regularization

3.1 Introduction

In a capacitated clustering problem (CCP), a set of n nodes must be partitioned into p disjoint clusters so that the total dissimilarity within each cluster is minimized, and constraints on maximum cluster capacities are obeyed. The CCP has a wide range of real-world applications. For example, when designing a distribution network, a set of customers must be supplied goods from warehouses subject to the capacity of warehouses. In the topological design of computer communication networks, the network nodes need to be divided into groups, and a concentrator location must be selected for each group so that all the nodes in a group can be assigned to the same concentrator without violating capacity constraints (Pirkul, 1987). Recently, the CCP has also been applied to genetics and population biology to solve the sibling reconstruction problem (Chou et al., 2012). In addition, many important applications involve solving the capacitated clustering as a sub-problem,

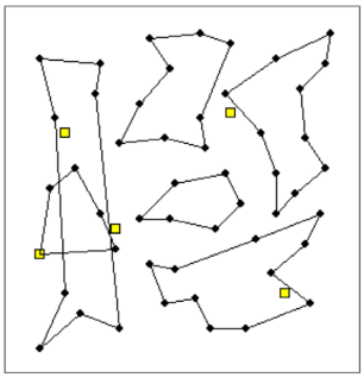
e.g. market segmentation, vehicle routing, and location-routing problems. In this paper, we provide a new perspective to this classic operations research problem through model-based capacitated clustering. Our algorithm postulate a statistical model for the points to be clustered, and takes care of the capacity constraints by adapting a recently-developed machine learning framework.

The CCP has been extensively studied in the operations research literature. The CCP can be formulated using integer programming models. Pirkul (1987) proposed an exact algorithms using a branch and bound with Lagrangean relaxation on the partitioning constraints. Baldacci et al. (2002) presented a set partitioning approach. Because the CCP problem is NP-hard (Mulvey and Beck, 1984), exact approaches oftern cannot find the optimal solution given limited time for practical-sized problems.

Alternatively, meta-heuristics can be used to solve the CCP problem. Meta-heuristics are solution methods that combine local improvement procedures with higher level strategies to overcome the trap of local optimality (Glover and Kochenberger, 2003). Meta-heuristics that are used to solve CCP include genetic algorithms (Correa et al., 2004), variable neighborhood search (Fleszar and Hindi, 2008), tabu search (Bozkaya et al., 2003), scatter search (Scheuerer and Wendolsky, 2006), path relinking (Díaz and Fernandez, 2006), among others.

Almost all meta-heuristics begin with constructing an initial solution and improving upon it. A natural candidate for the construction stage could be a regular clustering algorithm such as k-means or hierarchical clustering. However when using these methods in the construction stage, a key challenge is to incorporate problem-specific knowledge such as capacity constraints. For example, in a capacitated vehicle-routing problem (CVRP), the vehicles have a given capacity limitation, and each point (generally a pickup or delivery location) has a given space and/or weight requirements.

Figure 1: Biased effect when using capacity as a stopping criterion (Barreto et al., 2007)



Problem-specific knowledge can be incorporated into the algorithm in an ad-hoc manner. A naive algorithm could avoid merging two clusters or stop adding points to a cluster if such an operation would violate the capacity constraints. The problem with this approach is that points naturally close to each other may be prevented from being grouped together because of the capacity constraint, while points that are far away may be forced into the same cluster. Barreto et al. (2007) demonstrated this undesired effect of adding a capacity constraint to an agglomerative hierarchical method in the setting of a capacitated location-routing problem (Figure 1). The "biased effect" of merging far away groups is evident in the two cluster on the left side of Figure 1. Another approach is to ignore the capacity constraint when first forming the cluster and then later swap points between clusters. Such swap procedures can be very time consuming because one needs to check feasibility for every possible exchange.

Therefore, different construction heuristics have been proposed for the CCP, most of which adapts the idea of k-means (or k-median) clustering to the capacitated setting. In a k-means clustering, random points are chosen as the initial cluster centers, then the algorithm iterates between two steps: 1) assigning the points to the current cluster centers 2) re-computing new cluster centers to minimize with-in cluster dissimilarity. Construction heuristics for CCP focus on changing the

first step in-order to minimize the baised effect demonstrated in Figure 1. For example, the classic Mulvey and Beck (1984) algorithm would assign the points to current set of centers, as long as the capacity constraint is not violated, in the descending order of "regret". The regret is defined as the distance between the closest and the second closest centers.

The resemblance between these CCP construction heuristics and traditional clustering methods motivates this research. As we know, the k-means clustering belongs to the broader category of unsupervised learning in the domain of statistical machine learning, which means that there is no additional constraint on assignment of observations to clusters, and we simply let the data dictate the best cluster for each observation. Beyond the simple k-means, there is a variety of other unsupervised learning methods that are developed in statistics and machine learning field, one of which is the model-based clustering (Banfield and Raftery, 1993; Fraley and Raftery, 2002). Based on probability models, it offers a more principled and flexible alternative to heuristic algorithms such as k-means. This research translates the model-based clustering to a CCP setting, and benchmarks the performance of our model-based CCP heuristics. Table 1 positions our research relevant to other statistical and heuristic clustering methods.

Table 1: Overview of Model-based CCP

		Method	
		Non-parametric clustering	**Model-based clustering**
Resource	**Uncapacitated**	k-means, hierarchical clustering	Gaussian mixture models
	Capacitated	Mulvey and Beck (1984) Ahmadi and Osman (2004)	*Model-based CCP (this paper)*

In particular, we propose solving the CCP from a statistical point of view using expectation maximization (EM) (Dempster et al., 1977) on Gaussian mixture models. The EM algorithm provides an iterative approach to maximize the likelihood of finite mixture models with latent variables

and is regarded as an effective data mining algorithm (Wu et al., 2008). While EM has been widely used in statistics, computer science, and marketing research, it has had very limited usage in operations research, partially due to the difficulty of incorporating external constraints. Our strategy for building in the capacity constraint involves using semi-supervised machine learning, namely posterior regularization (PR). Proposed by Ganchev et al. (2010), the PR framework for latent variable models arises from a paradigm called weakly supervised learning. It allows prior knowledge to be introduced into models that are traditionally considered as unsupervised learning. The authors show that prior knowledge can be encoded as constraints on posterior probabilities and can be used to guide the outputs on various tasks in natural language processing such as part-of-speech tagging, word alignment and dependency grammar parsing.

We seek to adapt the PR framework to the capacitated p-median problem (CPMP), one of the most-studied variations of the CCP. We propose a principled way to introduce the capacity constraints in an EM algorithm, and we show that the method can produce high quality feasible or near-feasible solutions. We are able to identify promising initial solutions for the CPMP by embedding the capacity constraint into the process of forming clusters. The advantages of our algorithm include the following. (1) It is relatively easy to implement since it is based on EM, one of the most popular methods in statistical machine learning. (2) Our empirical results show that it has better performance than extant construction heuristics, and has even superior performance in stochastic problems and when the points have a clustered point pattern distribution.

We view our contribution as twofold: (1) We bring statistical methods that can have a role in solving discrete-optimization problems to the attention of the OR community, and we motivate researchers to revisit classical problems from a probabilistic perspective. (2) We show that EM combined with posterior regularization is a promising method for the CCP that provides results

comparable to other heuristics.

The structure of the paper is as follows. In Section 2 we introduce the CPMP as a mixed-integer linear programming (MILP) model. In Section 3 we briefly introduce the Gaussian mixture model and the EM algorithm to maximize its likelihood. We also discuss the parsimonious variations of the Gaussian mixture model, which we will use to solve the CPMP. In Section 4 we describe the PR framework, and we discuss how it can be adapted to solve the CPMP. In Section 5 we present the computational results. In Section 6 we evaluate the performance of a GRASP extension of the algorithm. Lastly we present our conclusions in the final section.

3.2 The Model

A capacitated p-median problem (CPMP) can be written as the following MILP model:

$$\text{minimize} \quad \sum_i \sum_j D_{ij} x_{ij} \tag{1}$$

$$\text{subject to} \quad \sum_j x_{ij} = 1 \quad \forall i, \tag{2}$$

$$\sum_i d_i x_{ij} \leq y_j C \quad \forall j, \tag{3}$$

$$\sum_j y_j = k, \tag{4}$$

$$x_{ij}, y_j \in \{0, 1\}, \ \forall i, j; \tag{5}$$

where

- $i = 1 \ldots n$ is the index of points to allocate and also of possible medians, where k medians will be located;
- $j = 1 \ldots n$ is the index of all possible cluster centers or medians;

- d_i is the demand of each point i and C is the capacity of each possible cluster;
- D_{ij} is the distance from point i to median j ;
- y_j are binary variables, with $y_j = 1$ if point y is selected to be a cluster median;
- x_{ij} are binary variables, with $x_{ij} = 1$ if point i is assigned to median j and $x_{ij} = 0$ otherwise;
- The objective of CPMP (1) is to minimize the sum of distance from points to the cluster medians. Constraint (2) ensures that all points are allocated to exactly one cluster median. Constraint (3) imposes the constraints on cluster capacities, and constraint (4) set the number of medians to k, while constraint (5) enforces the binary conditions.

3.3 Model-based Clustering with EM Algorithm

Cluster analysis is used to detect groups in a set of objects such that the members within each cluster are similar to each other. As we discussed earlier, clustering algorithms can be divided into two types: non-parametric clustering algorithms such as k-means, and EM algorithms that fit a Gaussian mixture model. In a model-based clustering approach, the objective is no longer minimizing an objective function (such as total within cluster distance), but rather maximizing the likelihood of a probability model. Here we briefly describe the general Gaussian model-based clustering method using EM and its two parsimonious variations. The models in this section do not take demand constraints into consideration.

3.3.1 Gaussian Mixture Model with EM

Let $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}$ be a set of observations in $\mathbb{R}^d$ that arise from a mixture of k groups. The probability that an observation comes from the jth mixture component is π_j, where $0 \leq \pi_j \leq 1$ and $\sum_{j=1}^{k} \pi_j = 1$. The mixture density is

$$f(\boldsymbol{x}) = \sum_{j=1}^{k} \pi_j \Phi(\boldsymbol{x}|\mu_j, \Sigma_j), \tag{6}$$

where $\Phi(\boldsymbol{x}|\mu_j, \Sigma_j)$ denotes a Gaussian density with mean μ and variance matrix Σ. Below we summarize the notations used to describe the Gaussian mixture model and EM algorithm.

- Index of mixtures: $j = 1, \ldots, k$,
- Index of observations: $i = 1, \ldots, n$,
- Dimension of observations: d,
- Observations: $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}$,
- Mixture probabilities: $\pi_1, \ldots, \pi_k$,
- Probability that observation i arises from jth mixture : $p_j^{(i)}$.

The EM algorithm (Dempster et al., 1977) is an iterative method to compute the maximum likelihood estimation for probability models with missing or latent data. In the context of mixture models, the observed data are the $\boldsymbol{x}^{(i)}$, and the latent part of the data is $z_j^{(i)}$ with its value equal to 1 if $\boldsymbol{x}^{(i)}$ belongs to mixture j, and 0 otherwise. The log likelihood of the complete model is then

$$l(\pi, \mu, \Sigma, z) = \sum_{i=1}^{n} \log(\sum_{j=1}^{k} \pi_j \Phi(\boldsymbol{x}|\mu_j, \Sigma_j). \tag{7}$$

The EM algorithm alternates between the E-step and the M-step. The E-step computes the conditional expectation of the latent variables given the current value of the parameter estimates:

$$\begin{aligned} p_j^{(i)} &= E(z_j^{(i)}|\boldsymbol{x}^{(i)}; \pi, \mu, \Sigma) = p(z^{(i)} = j|\boldsymbol{x}^{(i)}; \pi, \mu, \Sigma). \\ &= \frac{p(\boldsymbol{x}^{(i)}|z^{(i)} = j; \mu, \Sigma) p(z^{(i)} = j; \pi)}{\sum_{l=1}^{k} p(\boldsymbol{x}^{(i)}|z^{(i)} = l; \mu, \Sigma) p(z^{(i)} = l; \pi)}. \end{aligned} \tag{8}$$

We can find $p(\boldsymbol{x}^{(i)}|z^{(i)} = j; \mu, \Sigma)$ by evaluating a multivariate Gaussian density with mean $\boldsymbol{\mu}_j$ and covariance $\boldsymbol{\Sigma}_j$ at point $\boldsymbol{x}^{(i)}$. The current estimate of mixture probability π_j gives us $p(z^{(i)} = j; \pi)$.

In the M-step, the model parameters are updated:

$$\pi_j \leftarrow \frac{1}{n} \sum_{i=1}^{n} p_j^{(i)}, \tag{9}$$

$$\boldsymbol{\mu}_j \leftarrow \sum_{i=1}^{n} p_j^{(i)} \boldsymbol{x}^{(i)} / \sum_{i=1}^{n} p_j^{(i)}, \tag{10}$$

$$\boldsymbol{\Sigma}_j \leftarrow \sum_{i=1}^{n} p_j^{(i)} (\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)/(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_j)^T / \sum_{i=1}^{n} p_j^{(i)}. \tag{11}$$

The k-means algorithm can be thought of as a simpler non-probabilistic alternative to Gaussian mixtures. Despite having no explicit notion of cluster covariances, one can also view the k-means as a special case of the Gaussian mixture clustering, if one were to: (a) fix a priori all the covariances for the k components to be the identity matrix (and not update them during the M-step), and (b) during the E-step, for each data vector, assign a membership probability of 1 for the component it is most likely to belong to, and 0 for all the other memberships (in effect make a "hard decision" on component membership at each iteration).

3.3.2 Why Model-Based Clustering?

Model-based clustering provides a principled and flexible way to conduct clustering analysis. The flexibility comes from the fact that popular clustering heuristics are approximate methods for a certain model. For example, k-means and Ward's method maximizethe Gausian likelihood when the covariance matrix is the same multiple of the identity matrix across mixtures (Fraley and Raftery, 2002). As another example, Dasgupta and Raftery (1998) show that model-based clustering can be extended to detect irregular shapes such as the parallel rectangles and arrow shapes. Although not a focus in this paper, one can imagine that these structures can also be meaningful when taking capacity into consideration.

In terms of performance, Yeung et al. (2001) show that model-based clustering achieves better results than heuristic-based clustering for gene expression data. In addition, the model can provide insights into when a clustering method may perform better (Bock, 1996).

3.3.3 Parsimonious Models

A practical issue with multivariate normal models is that the number of parameters can grow rapidly with the number of clusters. For example, in the well-known facility-location problem with two dimensions, a k-cluster full normal mixture model will have $(\text{number of mean parameters}) + (\text{number of covariance parameters}) + (\text{number of mixture proportion parameters}) = 2k + 3k + (k - 1) = 7k - 1$ parameters. Too many parameters compared to the number of data points can result in issues such as degradation of performance and under-specified models (Raftery and Dean, 2006). In particular, the estimation for a full covariance matrix will be singular or near singular.

Banfield and Raftery (1993) show that the covariance matrix can be parameterized in terms of its eigenvalue decomposition in the form

$$\Sigma_g = \lambda_g \boldsymbol{D}_g \boldsymbol{A}_g \boldsymbol{D}_g^T. \tag{12}$$

This approach was later generalized by Celeux and Govaert (1995) into Gaussian parsimonious clustering models to impose various restrictions on covariance matrices of the distribution. We specifically consider the following two parsimonious models, since neither the full model nor other parsimonious models are able to give satisfactory performance in our study.

1. **EII:** $\Sigma_1 = \ldots = \Sigma_k = \sigma^2 I$. In Model EII, σ^2 is the only unknown covariance parameter.

2. **VII:** $\Sigma_1 = \ldots = \Sigma_k = \sigma_1^2 I, \ldots, \sigma_k^2 I$. In Model VII, the number of covariance parameters is equal to the number of clusters.

Imposing these restrictions on the covariance matrices essentially restricts the shape, volume, or orientation of each cluster. For example, in Model EII each cluster has a spherical shape and equal volume; while in Model VII the clusters are still spherical, but the volumes are allowed to vary.

Another consequence of using a more restricted covariance structure is that in the M-step of the EM algorithm, the inference for covariance parameters becomes simpler. Instead of iterative optimization procedures which are required for the full model, many parsimonious models have closed-form solutions for Σ in the M-step. In Model EII, the updated σ^2 is

$$\sigma^2 = \frac{\text{tr}(W)}{nd}, \tag{13}$$

where $W = \sum_{j=1}^{k} \sum_{i=1}^{n} p_j^{(i)} (\boldsymbol{x}^{(i)} - \bar{\boldsymbol{x}}_j)(\boldsymbol{x}^{(i)} - \bar{\boldsymbol{x}}_j)'$.

In the M-step of Model VII, σ_j^2 can be calculated as

$$\sigma_j^2 = \frac{\text{tr}(W_j)}{n_j d}, \tag{14}$$

where $W_j = \sum_{i=1}^{n} p_j^{(i)} (\boldsymbol{x}^{(i)} - \bar{\boldsymbol{x}}_j)(\boldsymbol{x}^{(i)} - \bar{\boldsymbol{x}}_j)'$ and $n_j = \sum_{i=1}^{n} p_j^{(i)}$.

3.3.4 Posterior Regularization (PR) Framework

3.3.4.1 General Description

In this section, we introduce the recently-developed PR framework, which offers the key insight for us to incorporate demand constraints into the model-based clustering. In the PR framework, the constraints serve as indirect supervision to the probabilistic learning framework. The posterior probability distributions of latent variables are guided toward desired behavior. To promote algorithm efficiency, we must define the set of valid posterior distributions with respect to expectation of constraints. That is, suppose Q is a set of all valid distributions, $q(\boldsymbol{Z})$ is a distribution of latent variables $\boldsymbol{Z}$, and $\boldsymbol{\phi}(\boldsymbol{X}, \boldsymbol{Z})$ is a function of observed variables $\boldsymbol{X}$ and latent variables $\boldsymbol{Z}$. The desired set of posterior distributions is

$$Q = \{q(\boldsymbol{Z}) : \boldsymbol{E}_q[\boldsymbol{\phi}(\boldsymbol{X}, \boldsymbol{Z})] \leq \boldsymbol{b}\}. \tag{15}$$

In order to find the $q(\boldsymbol{Z})$ within the above region such that the overall model likelihood is maximized, Ganchev et al. (2010) show that at each of the E-step of the EM algorithm we solve the following optimization problem:

$$\min_{q} \boldsymbol{KL}(q||p_\theta(\boldsymbol{z}|\boldsymbol{x})) \qquad \text{s.t. } \boldsymbol{E}_q[\boldsymbol{\phi}(\boldsymbol{X}, \boldsymbol{Z})] \leq \boldsymbol{b}, \tag{16}$$

where $\boldsymbol{KL}$ is the Kullback–Leibler divergence, which is a measure of the difference between two distributions. It is defined as $\boldsymbol{KL}(q||p) = \sum_q q \log(q/p)$. The M-step remains unchanged.

The above problem can be solved more efficiently in its dual form

$$\max_{\boldsymbol{\lambda} \geq 0} -\boldsymbol{b} \cdot \boldsymbol{\lambda} - \log Z(\boldsymbol{\lambda}), \tag{17}$$

where $Z(\boldsymbol{\lambda}) = \sum_{\boldsymbol{Z}} p_\theta(\boldsymbol{Z}|\boldsymbol{X}) \exp[-\boldsymbol{\lambda}^* \cdot \phi(\boldsymbol{X}, \boldsymbol{Z})]$, and the solution to the primal is given by

$$q^*(\boldsymbol{Z}) = \frac{p_\theta(\boldsymbol{Z}|\boldsymbol{X}) \exp[-\boldsymbol{\lambda}^* \cdot \phi(\boldsymbol{X}, \boldsymbol{Z})]}{Z(\lambda^*)}. \tag{18}$$

Notice that in a regular EM algorithm without PR, Eq.(16) is a problem of minimizing the objective function without the expectation constraint. In other words, at the E-step of the regular EM algorithm, we are solving $\min_q \boldsymbol{KL}(q||p_\theta(\boldsymbol{z}|\boldsymbol{x}))$. The optimal solution for $q(\boldsymbol{Z})$ is $p_\theta(\boldsymbol{Z}|\boldsymbol{X})$, which is the posterior probability of latent variables given the current parameters and the observed variables. In the PR framework, we instead restrict q to the set Q defined in Eq.(15). The restriction trades off a smaller maximum lower bound of likelihood for desired posteriors. Therefore, a simple way of interpreting the PR framework is to add a penalty term $\exp[-\boldsymbol{\lambda}^* \cdot \phi(\boldsymbol{X}, \boldsymbol{Z})]$ to tune down the posterior probability in the E-step of the EM algorithm for violations of constraints.

3.3.4.2 Posterior Regularization Framework for Capacitated Clustering

The constraints are defined in terms of the expectation of $\boldsymbol{q}$, which is a distribution of latent variables

$$\boldsymbol{E}_q[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{z})] \leq \boldsymbol{b}. \tag{19}$$

For a capacitated clustering problem, if we let $z_j^{(i)} = 1$ represent the event that point i is assigned to cluster j, the problem-specific knowledge we want to include when forming clusters is

$$\sum_i z_j^{(i)} d_i \leq C, \ \ \forall j. \tag{20}$$

Constraint (20) states that the total capacity in each cluster should not exceed C. Expressing the above constraint as posterior constraints in an EM-algorithm, we have

$$E_q[\sum_i (z_j^{(i)} d_i)] \leq C, \ \ \forall j. \tag{21}$$

Computationally, the optimization problem is solved by maximizing the dual at each E-step. The dual problem is

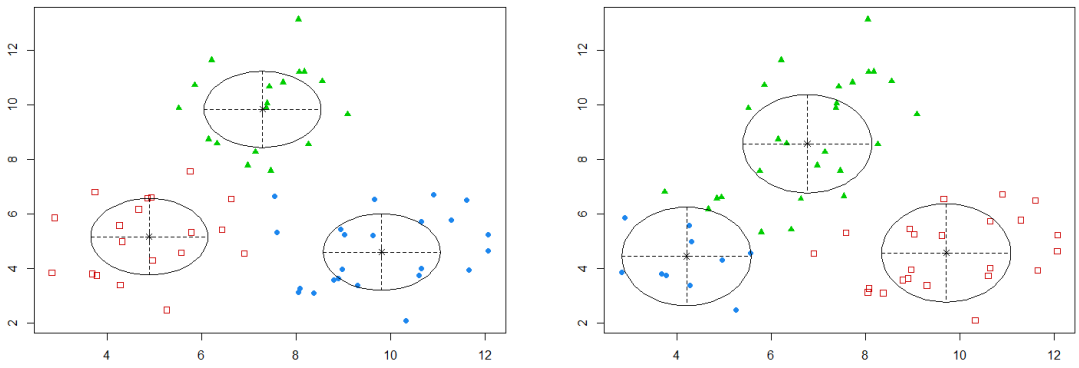
$$\max_{\lambda_1 \cdots \lambda_k \geq 0} -\sum_{j=1}^{k} C\lambda_j - \log\Big(\sum_{j=1}^{k}\big(\prod_{i=1}^{n} p_j^{(i)} exp(-\sum_{j=1}^{k}\lambda_j \sum_{i=1}^{n} d_i p_j^{(i)})\big)\Big). \tag{22}$$

The maximization problem can be easily solved using standard nonlinear optimization methods such as the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method with barrier. The primal solution is given by

$$q_j^{(i)} = p_j^{(i)} exp(-\lambda_j \sum_{i=1}^{n} d_i p_j^{(i)})/Z, \tag{23}$$

where $Z = \sum_{j=1}^{k} q_j^{(i)}$, which is a normalization factor to ensure that q is a valid probability distribution over each point i.

Figure 2: An illustrative example of capacitated clustering based on PR framework



(a) Uncapacitated EM

(b) Capacitated EM

3.3.4.3 An Illustrative Example

To illustrate the effect of posterior regularization, we simulate three 20-points groups; for each group of points, their (x, y) coordinates follow a bivariate normal distribution. The points in the first group centered at $(7.5, 10)$ and the second group centered at $(10, 5)$ have demand of 0.5; the points in the third group centered at $(5, 5)$ have demand of 2.

Figure 2 (a) shows the clustering result of the regular EM algorithm. As expected the three groups with about equal number of points are formed according to their (x, y) coordinates. Suppose the capacity constraint of each cluster is set at 20, then obviously the group at the bottom-left corner will violate the capacity constraint. The capacitated clustering result using posterior regularization is demonstrated in Figure 2 (b). We are able to attain the clusters that respect the capacity constraints while having points in a group naturally close to each other.

3.4 Heuristic Algorithm Based on PR Framework

Given the adaptation of PR Framework described in Section 3.3.4.2, we need several adjustments for practical considerations.

3.4.1 Penalizing Posterior Distribution

First of all, we note that the number of dual decision variables, λ_j, equals the number of desired clusters k in (22). Thus solving a problem with k clusters will require solving a non-linear optimization problem with k decision variables at each iteration of EM. Second, we notice that during the initial few iterations of EM, when there are relatively large capacity violations in some clusters, the optimal λ of the dual problems will be very large as well. Because EM converges to a local maxima of log likelihood, this may cause many clusters to be empty in the final solution. In other words, adding capacity constraints can result in the increasing number of undesired local optima.

To get around the above two issues, we propose that instead of actually solving (22) in the E-step, we simply check if (21) is satisfied for the current posterior probability matrix $P(Z|X)$. More specifically, given the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ obtained from the last M-step, we calculate the posterior probability $p_j^{(i)}$ as in E-step of regular EM (without PR). Then for each column j of the probability matrix we calculate $\sum_i d_i p_j^{(i)}$, and if the quantity is greater than the capacity constraint C, we apply a penalty to the column by multiplying it by a penalty coefficient r, where $0 < r < 1$, that is

$$p_j^{(i)} \leftarrow r p_j^{(i)} \quad \text{if} \quad \sum_i d_i p_j^{(i)} \geq C, \quad \forall j. \tag{24}$$

Lastly, to make sure that each row of the probability matrix is a valid marginal distribution, we normalize them so that the sum of each row is 1.

While this adjustment results in much faster computation and more stable results, it nullifies the guaranteed likelihood increase of the EM algorithm with PR at each iteration. Fortunately, this is less of an issue practically. As shown in Fig 3, the log likelihood of the model still has a consistently increasing trend. Our empirical evaluation shows that the final results are very sensible despite the non-guaranteed likelihood convergence. After a number of iterations, the log likelihood will vary within a small region. This indicates that eventually the EM algorithm will oscillate between several promising solutions when trying to balance between satisfying the constraints and further increasing the model likelihood. Therefore, we implement a simple convergence check by calculating the log likelihood after every 5 iterations, and check if the standard deviation of the last 10 log likelihoods is smaller than certain threshold ϵ. We set $\epsilon = 1$ for our experiments.

3.4.2 The Initialization of EM

Researchers have proposed different initialization strategies for CPMP heuristics. For example, Mulvey and Beck (1984) initializes with random nodes as centers. Osman and Christofides (1994) propose a step to find a set initial centers that are spread out. Ahmadi and Osman (2004) and Osman and Ahmadi (2007) use a density based approach to locate the most promising centers.

In our probabilistic model, there is no set of initial centers *per se*. Instead, we can either specify the initial conditional probability matrix (the probabilities of nodes belonging to clusters) or the initial mean vectors and covariance matrices in the context of Gaussian mixture models. We adopt a simple random initialization where the mixing proportions are generated from a symmetric Dirichlet distribution. Besides simplicity of implementation, there are several reasons for us to choose the strategy: 1) it is considered as the standard, and most frequently used initialization strategy for mixture models (Karlis and Xekalaki, 2003); 2) a comprehensive numerical study by

Biernacki et al. (2003) demonstrates that under low dimension, other more sophisticated strategies offer no significant improvement; and 3) visual inspection of final solutions produced by our heuristic shows that the quality of solutions depends more on the assignment of nodes, as the clusters are reasonably spread out.

3.4.3 The Assignment of Nodes

After numbers of iterations between the regularized E-step and the M-step, we have a posterior distribution of the latent variable $p_\theta(\boldsymbol{Z}|\boldsymbol{X})$ along with the parameters $\boldsymbol{\mu}_j$ and $\boldsymbol{\sigma}_j$. For CPMP, we now need to consider assigning the observed nodes X to the cluster medians. Choosing the cluster medians is straight forward, as we can simply pick the k nodes that are closest to the cluster means $\boldsymbol{\mu}$.

We investigate several methods to assign the nodes to the medians. In the CPMP literature, orders of node assignment include: increasing or decreasing order of demand, increasing order of distances from nodes to medians, etc. Using the decreasing order of regret values is a popular choice; however the definitions of regret differ based on the implementation.

Let $O = o_1, \ldots, o_k$ be the set of medians. For every node x_i, Mulvey and Beck (1984) define regret $R(x_i)$ as the distance between the closest median o_{i1} and the second closest median o_{i2}:

$$R(x_i) = d(o_{i1}, o_{i2}), \tag{25}$$

Ahmadi and Osman (2004) define the regret as the savings of assigning node i to the closest median compared with assigning it to the second closest median:

$$R(x_i) = d(a_i, o_{i1}) - d(a_i, o_{i2}). \tag{26}$$

Similarly, we define our regret function as the difference between the posterior probabilities of the two clusters with the highest probabilities of generating x_i:

$$R(x_i) = p_\theta(z_1|x_i) - p_\theta(z_2|x_i). \tag{27}$$

Note that, the regret function based on the posterior probabilities not only considers the location of medians (as in Mulvey and Beck (1984)), and the relative location between nodes and medians (as in Ahmadi and Osman (2004)), but also the entire probabilistic model. This means that the capacity constraint of all clusters is implicitly considered in addition to the geographic locations.

3.4.4 Local Search Strategies

After a predetermined number of EM iterations, improvements to the solution are attempted through local search. Our algorithm explores two different local search neighborhoods: Shift and Swap. The Shift neighborhood contains the solutions generated from shifting one point assigned to one median to another median. The Swap neighborhood contains all pairwise interchanges of non-median nodes between clusters. For a given solution, If a certain Shift or Swap operation will improve the current solution, and at the same time not violate capacity constraints, the operation can be made.

We also evaluate two selection strategies for choosing a solution in the neighborhood. The first-improvement chooses the next candidate solution as soon as a feasible improvement is found in the neighborhood. The best-improvement evaluates all solutions within the neighborhood and accepts the one that gives the best improvement. After performing all feasible interchanges the solution is recorded. From our experience the best-improvement strategy gives better results without sacrificing much computing time, and is therefore used in subsequent studies. The local search

process terminates when no improving move can be found.

Algorithm 1 formally describes the heuristic based on the PR framework that can be used to solve a CPMP.

Algorithm 1 EM Clustering for Capacitated P-Median

Input: coordinates of points $\boldsymbol{x}^{(i)}, (i = 1, \ldots, n)$ with demand d_i, number of clusters k, cluster capacity C
Output: A set of k clusters medians; assignments from n points to medians.
Parameters: penalization constant r

Step 1. Initialization of EM
Initialize a $n \times k$ matrix P with entries denoted as $p_j^{(i)}$
Draw each row from a symmetric Dirichlet distribution $(\alpha = 1)$.

Step 2. EM Iterations
while not convergence **do**
  for all cluster j **do** ▷ Regular M-Step
    Update mixture parameters:
    $\pi_j \leftarrow \frac{1}{n} \sum_{i=1}^{n} p_j^{(i)}$,
    $\boldsymbol{\mu}_j \leftarrow \sum_{i=1}^{n} p_j^{(i)} \boldsymbol{x}^{(i)} / \sum_{i=1}^{n} p_j^{(i)}$,
    Update $\boldsymbol{\Sigma}_j$ according to Eq. (13) or (14).
  end for

  for all point i **do** ▷ Regular E-Step
    Update conditional probabilities:
    Compute $p_j^{(i)} \leftarrow p(z^{(i)} = j | \boldsymbol{x}^{(i)}; \boldsymbol{\phi}, \boldsymbol{\mu}_j, \sum_j)$ using Eq. (8).
  end for

  for all cluster j **do** ▷ Posterior Regularization
    if $\sum_i d_i p_j^{(i)} > C$ **then**
      $\boldsymbol{p}_j \leftarrow r \boldsymbol{p}_j$ (multiply column j of P by r).
    end if
  end for
  for all point i **do**
    $p_j^{(i)} \leftarrow p_j^{(i)} / \sum_j p_j^{(i)}$ (normalize each row of P to a probability distribution).
  end for
end while

Step 3. Determine Cluster Medians
for all cluster j **do**
Let $i = \arg\min_{i \in 1 \ldots n} d(\boldsymbol{x}^{(i)}, \boldsymbol{\mu}_j)$,
assign point i as median for cluster j.
end for

Step 4. Cluster Assignment
for all point i **do**
Let $R_i \leftarrow p^{(i)}_{(1)} - p^{(i)}_{(2)}$,
where $p^{(i)}_{(1)}$, $p^{(i)}_{(2)}$ are the largest and second largest entries in row i of P.
end for

for all decreasing $i \in R_i$ **do**
while point i is not assigned or infeasible **do**
$j' \leftarrow \arg\max_{j \in 1 \ldots k} p^{(i)}_j$
if $Demand(j') + Demand(i) \leq C$ **then**
Assign point i to cluster $\arg\max_{j \in 1 \ldots k} p^{(i)}_j$.
else
Set $p^{(i)}_j = 0$
end if
end while
end for

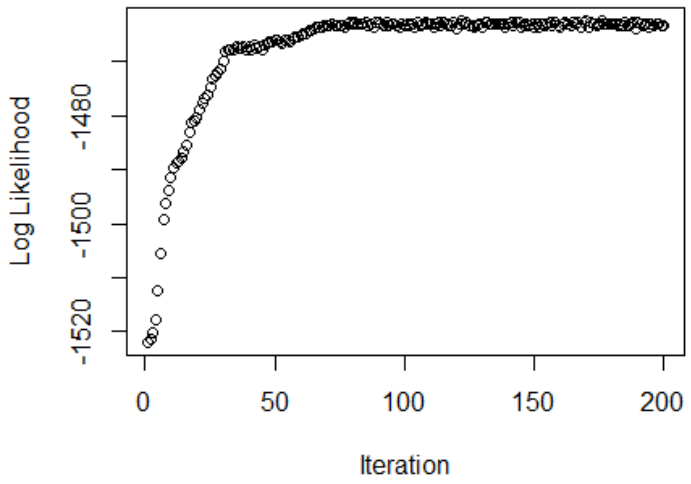
Step 5. Local Search
while improvement **do**
Shift one points from one median to another if it improves the solution the most among all feasible shifts.
Swap allocation of two points from different medians if it improves the solution the most among all feasible swaps.
end while

3.5 Computational Results

3.5.1 Analyses with Test Instances

We test our suggested algorithm's performance using the following CPMP problem instances. The 20 instances coded as $p1$ to $p20$ are from Osman and Christofides (1994). The first 10 instances have $n = 50$ and $k = 5$, and the other 10 instances have $n = 100$ and $k = 10$. Our heuristics

Figure 3: Log likelihood in EM with posterior regularization heuristic



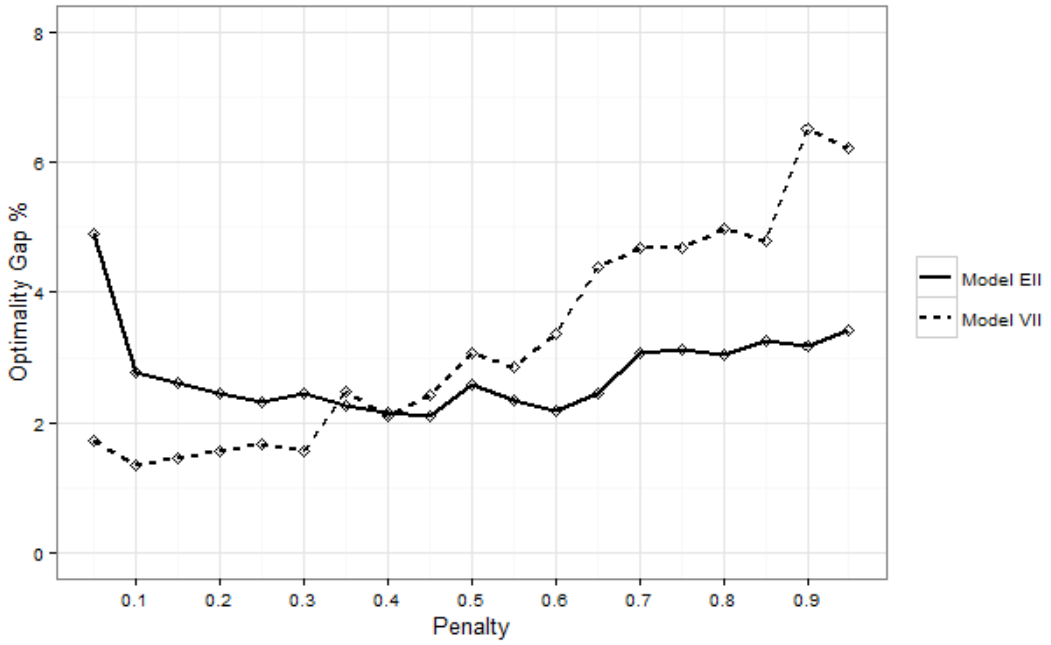
are coded in Java and all tests are performed on an Intel i5-3570k processor under the Microsoft Windows 8 operating system.

3.5.1.1 Effect of Algorithm Parameters

With the test instances, we first study the effects of the posterior penalty parameter r and the parsimonious model on the performance. Since the model choice and posterior penalty is mostly influential on the solution construction stage, we report the average optimality gap from the solutions before the local search stage. The optimal values are computed by solving the integer programming model with Gurobi MILP solver.

It is evident from Figure 4 that Model VII offers higher quality solutions than Model EII for the test instances. This is not surprising, since Model VII explicitly parameterizes cluster size to be different, and therefore has the potential to perform better when node demands are more heterogeneous. Also, a posterior penalty r between 0.1 and 0.3 seems to offer the best results for

Figure 4: Effect of different posterior penalty parameters and Gaussian parsimonious models



VII Models. We choose Model VII with penalty parameter $r = 0.1$ for subsequent studies.

3.5.1.2 Comparing with Model-free CCP

Table 1 provides results obtained by comparing model-based CCP (EM) with other model-free methods. We implement the Mulvey-Beck (MB) heuristic (Mulvey and Beck, 1984) on the same computing platform and with the same local search procedures. For the model-based CCP and the MB heuristics we report the best results from 40 runs, which is a standard strategy for randomly initialized clustering method. We also compare the performance with other methods' performances found in literature. The density search constructive method (DSCM) is a method proposed by Ahmadi and Osman (2004), and uses a density function to find cluster centers and then uses a regret function to find assignments. HOC is the naive construction algorithm used in Osman and Christofides (1994).

Table 2 reports the computational results of the standard test instances. The results show that our heuristic is competitive compared to DSCM and MB, and therefore, provides initial validation for its application to the CPMP.

Table 2: Comparison of best solutions

	EM		MB (Implemented)		DSCM (reported)		HOC	Optimal
	Const.	LS	Const.	LS	Const.	LS		
1	713	713	713	713	713	713	786	713
2	749	740	740	740	740	740	816	740
3	770	754	779	764	758	753	972	751
4	656	651	651	651	651	651	891	651
5	674	674	696	666	666	666	804	664
6	786	778	820	787	783	778	882	778
7	792	792	811	788	787	787	968	787
8	847	822	846	838	872	839	945	820
9	724	718	718	717	724	724	752	715
10	847	829	841	838	837	837	1017	829
11	1033	1009	1026	1015	1006	1006	1761	1006
12	986	975	976	969	974	970	1567	966
13	1030	1026	1042	1026	1065	1056	1847	1026
14	989	983	1019	988	1009	1009	1635	982
15	1114	1096	1129	1105	1100	1099	1517	1091
16	971	956	973	958	983	979	1780	954
17	1036	1034	1071	1048	1124	1123	1665	1034
18	1089	1058	1088	1053	1073	1062	1345	1043
19	1071	1045	1077	1037	1066	1055	1634	1031
20	1063	1018	1107	1059	1053	1051	1872	1005
Avg. Gap (%)	1.920	0.465	2.918	0.933	2.072	1.594	41.575	

Note: Const. includes the solution generated from the construction stage. LS includes the solution values after a local search stage.

3.5.2 Point Pattern and Performance of Heuristics

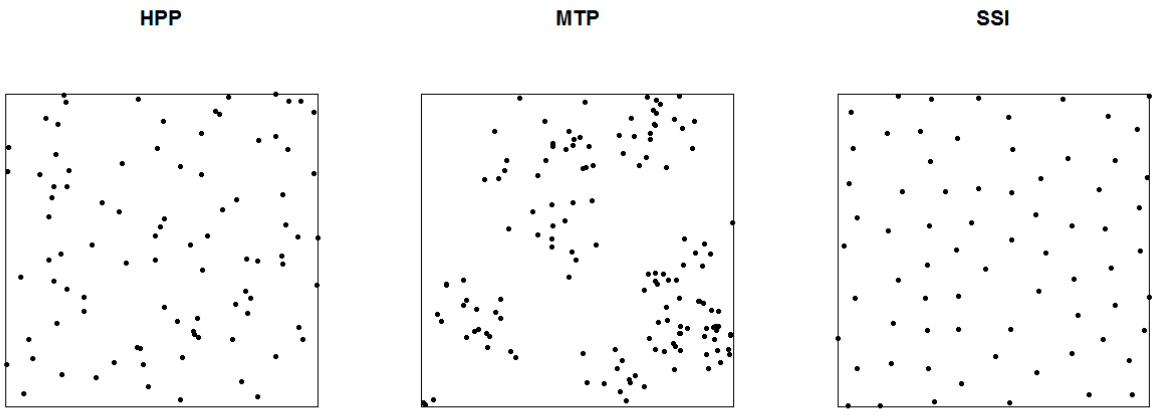
After confirming the effectiveness of model-based capacitated clustering with standard test instances, we next investigate the impact of spatial patterns on the effectiveness of model-based capacitated clustering. According to the definition of Hudson and Fowler (1966), pattern is the

characteristic of a set of points which describes the location of these points in terms of the relative distances and orientations of one point (or group of points) to another point (or group of points). Point pattern analysis (PPA) has become an important application in recent years, particularly in crime analysis, epidemiology, and facility-location planning and management (Boots and Getis, 1985). It is also an essential component of modern geographic information system (GIS) systems (Fotheringham and Rogerson, 2013). However the analysis of spatial patterns has received surprisingly little attention in the OR community, considering that Park (1989) showed that performance of locational decision making is systematically related of the spatial characteristics of the environment in which they are used. In the CCP literature, Mulvey and Beck (1984) and Osman and Ahmadi (2007) compared instances where (x, y)'s are uniformly distributed or drawn from a single normal distribution. To the best of our knowledge no study has formally investigated the statistical models of PPA, which we discuss the details below.

Intuitively, since we build our method based on Gaussian mixture models, we expect that if the spatial randomness of a CCP shares similar characteristics with a $2d$ Gaussian distribution, then the performance of our model-based approach should be better. Formally, a spatial point pattern (SPP) consists of the locations of a finite number of points in a region R^d, where the locations are modeled as d-dimensional random variables. We consider three classes of stochastic process to generate patterns.

- **Homogeneous Poisson Process (HPP)** The HPP is a stochastic mechanism for generating or describing SPPs. It is also known as complete spatial randomness (CSR). A homogeneous Poisson process is defined as: for some $\lambda > 0$, the number Y of events within the region S follows a Poisson distribution with mean $\lambda|S|$, where $|.|$ denotes a two-dimensional area.

Figure 5: Examples of different point patterns



- **Modified Thomas Process (MTP)** First described in Diggle et al. (1976), MTP can be used to generate points with natural clustering. It consists of three stages. Firstly, "parent" points are distributed randomly over the plane according to a Poisson process with density λ per unit area. Secondly, each parent independently produces a random number of "offspring" according to a Poisson distribution with mean μ. Lastly, the locations of these offspring are distributed according to the symmetric radial Gaussian with parameter σ.

- **Simple Sequential Inhibition (SSI)** This process can be used to describe points which are regular in pattern. The points are distributed in the area one-by-one. The distribution of each subsequent point is conditional on all previously realized points. More specifically, each new point is generated uniformly in the area, but the new point is rejected if it lies closer than r units from any existing point. The process terminates when desired number of points are generated.

Figure 5 represents the three basic types of point patterns. HPP generates complete spatial

randomness, while the other two show clustering and regularity patterns, respectively. For each of these patterns, we generate 30 test instances with number of nodes $n = 100$, capacity $C = 120$, and number of clusters $k = 10$. The tightness coefficients (total demand as a percentage of total capacity) are uniformly distributed from 0.6 to 0.8. Given a specified tightness coefficient, the demand for individual nodes are simulated from a symmetric Dirichlet distribution with $\alpha = 2$.

Table 3 summarizes the performance of capacitated EM and Mulvey-Beck heuristics. We observe significant performance advantages for capacitated EM across all point patterns. The difference is particularly obvious when nodes have a natural clustered pattern (as shown in Figure 5b). The average optimality gap is 4.07% when using Mulvey-Beck, and only 1.59% when using capacitated EM.

Table 3: Gap to optimal (%) under different point patterns

	HPP (random pattern)	MTP (clustered pattern)	SSI (regular pattern)
EM	1.93(1.42)	1.59(1.74)	1.88(0.68)
MB	2.51(1.53)	4.07(3.36)	2.96(1.23)
P-value (paired t-test)	< 0.001	0.023	< 0.001

The results highlight the need of PPA before applying location-based heuristics, as these heuristics may not provide similar results for different patterns. Methods such as the Quadrat Test (Besag and Diggle, 1977) or the G-function Test (Clark and Evans, 1954) can be used to determine the point pattern.

3.5.3 Stochastic CPMP

We now consider a variation of the CPMP in which the demand of nodes are uncertain. Compared to the deterministic CPMP in Section 2, the following modifications are made to the model.

- Demand at each node is a random variable with a known probability distribution.

- The assignments of cluster medians must be completed before actual demands become known.

- The objective is to minimize the expected total assignment cost.

We formulate the stochastic CPMP using a chance-constrained model. In chance-constrained programing (Charnes and Cooper, 1959), a deterministic linear constraint set $a^T x \leq b$ is replaced by a set of chance-constraints $Pr(a^T x \leq b) \geq 1 - \alpha$. The new constraint set represents the probability that the deterministic constraint set is satisfied, and α is the allowable probability for the violation.

In the chance-constrainted CPMP, we let the d_i's be independent random variables representing node i's demand, and α be the allowed probability that the cluster exceeds its capacity. All other parameters follow the definitions given in Section 2.

$$\text{minimize} \qquad \sum_i \sum_j D_{ij} x_{ij} \tag{28}$$

$$\text{subject to} \qquad \sum_j x_{ij} = 1 \quad \forall i, \tag{29}$$

$$Pr(\sum_i d_i x_{ij} \leq y_j C) \geq 1 - \alpha, \quad \forall j, \tag{30}$$

$$\sum_j y_j = k, \tag{31}$$

$$x_{ij}, y_j \in \{0, 1\}, \ \forall i, j. \tag{32}$$

Charnes and Cooper (1959) and researchers in stochastic vehicle routing (for example Gendreau et al. (1996)) have shown that chance-constrained models can be transformed into deterministic optimization models. However, the transformed deterministic model may no longer be linear

and therefore requires more effort to find exact solutions. We consider two cases: 1) demands are drawn from a Poisson distribution, and 2) demands are from a normal distribution. Both cases are also considered in Lin (2009) with the following differences: 1) in Lin's problem, cluster medians are not required to be located at one of the nodes; 2) the number of clusters is endogenous to the model; 3) Lin assumes that the capacity requirements are heterogeneous across clusters.

3.5.3.1 Poisson Demand

Suppose that the demand in node i follows an independent Poisson distribution with mean μ_i. Because the sum of independent Posson random variables is also Possion distributed, the chance constrained capacity constraint can be written in the following deterministic form,

$$\sum_{k=0}^{y_j C} e^{-\sum_i \mu_i x_{ij}} \frac{(\sum_i \mu_i x_{ij})^k}{k!} \geq 1-\alpha, \quad \forall j. \tag{33}$$

In order for the above constraint to be satisfied, we need to first find a Poisson random variable ω with mean $\hat{\mu}$ such that $Pr(\omega \leq C) \geq 1-\alpha$, and then let

$$\sum_i \mu_i x_{ij} \leq \hat{\mu} y_j, \quad \forall j. \tag{34}$$

For any stochastic CPMP problem, the first step needs to be done once via a binary search since the Poisson CDF is monotonically decreasing in terms of $\hat{\mu}$ (Lin, 2009). Since the above constraint is equivalent to the capacity constraint in a deterministic CPMP we omit the discussion of this trivial case.

3.5.3.2 Normal Demand

Suppose that the demands are independent normally distributed, with mean μ_i and standard deviation σ_i for node i. In cluster j the total demand is normally distributed with mean $\sum_i \mu_i x_{ij}$ and

standard deviation $\sqrt{\sum_i \sigma_i^2 x_{ij}}$. The chance constraint is equivalent to the deterministic constraint,

$$\frac{y_j C - \sum_i \mu_i x_{ij}}{\sqrt{\sum_i \sigma_i^2 x_{ij}}} \geq z_{1-\alpha}, \quad \forall j, \tag{35}$$

which can be rewritten as

$$z_{1-\alpha}\sqrt{\sum_i \sigma_i^2 x_{ij}} + \sum_i \mu_i x_{ij} \leq C, \quad \forall j. \tag{36}$$

The deterministic model is now a non-quadratic mixed integer non-linear program that can cannot be solved using standard optimization packages. Fortunately, both the Mulvey-Beck heuristics and the EM algorithm can be adapted to solve the chance-constrained CPMP. For the Mulvey-Beck algorithm the change comes in each iteration when nodes are being assigned to current medians according the order of regrets. The feasibility check of whether cluster demand is exceeded after joining will be replaced by equation (36).

Similarly, for the EM based heuristic, the feasibility check in equation (24) can be replaced by

$$p_j^{(i)} \leftarrow r p_j^{(i)} \quad \text{if} \quad z_{1-\alpha}\sqrt{\sum_i \sigma_i^2 p_j^{(i)}} + \sum_i \mu_i p_j^{(i)} \leq C, \quad \forall j. \tag{37}$$

Table 4 shows the performace comparison between the adapted EM and Mulvey-Beck heuristic when we assume the standard test instances's demand follow a normal distribution with mean equal to the deterministic demand, and with a known standard deviation generated according to two levels of coefficients of variance, $cv = 0.05$ and $cv = 0.1$. We evaluate the cases when $\alpha = 0.02$ and $\alpha = 0.1$. For some instances the heuristics are not able to generate a feasible solution due to thightness of capacity constraints. For all other test instances EM almost always outperforms its non-parametric counterpart by a reasonable margin.

Table 4: Performance of heuristics on stochastic CPMPs

alpha	0.02				0.1			
cv	0.05		0.1		0.05		0.1	
Problem	EM	MB	EM	MB	EM	MB	EM	MB
1	721	724	738	762	726	746	726	746
2	740	748	748	748	740	740	755	748
3	784	825	796	856	761	832	784	828
4	657	680	657	679	655	675	655	692
5	683	742	-	768	674	729	683	742
6	782	825	919	1018	792	820	803	952
7	840	811	-	957	820	811	824	815
8	882	960	-	1054	860	962	882	936
9	762	778	-	-	727	729	-	-
10	-	-	-	-	851	1041	-	-
11	1063	1073	1091	1112	1035	1072	1057	1085
12	996	1005	1010	1007	997	1002	994	999
13	1035	1084	1064	1126	1027	1058	1035	1099
14	1026	1047	1040	1141	1012	1074	1020	1065
15	1152	1189	1182	1235	1129	1151	1157	1179
16	996	1022	1032	1124	991	978	990	1031
17	1063	1117	1104	1182	1060	1089	1099	1114
18	1176	1134	1178	1179	1114	1102	1174	1132
19	1085	1123	1159	1118	1111	1088	1096	1135
20	1237	1251	-	-	1122	1224	-	1299
Gap*	2.76%		4.22%		3.18%		3.91%	

*Gap is defined as the average of (MB-EM)/MB

3.6 Diversification Strategies via GRASP

In previous sections we use a multistart strategy by randomly initializing the EM. A profiling of our code suggests that although cold starting EM iterations each time can offer a high degree of diversification, such approach may not always be the most time-effective way to construct a solution. Therefore, we further investigate a greedy randomized adaptive search procedure (GRASP) based on the EM algorithm.

GRASP was first proposed by Feo and Resende (1989). Each iteration in the metaheuristic

consists of a construction phase and a local search phase. What differ GRASP from a conventional multi-start heuristic are the two features in its construction stage: Randomized Greedy, and Adaptive Selection. Instead of "throwing away" a set of P and μ produced by an EM convergence, the GRASP constructs multiple feasible solutions for the local search phase using a randomized greedy approach. There are two random components in the process: assigning nodes to clusters and deciding cluster centers.

We tested two randomization techniques when deciding which cluster a node should be assigned to. The first one is to use a standard restricted candidate list (RCL). The RCL includes feasible clusters corresponding to the top l Gaussian mixtures that have the highest probability of generating the node. Then the solution is constructed by randomly choosing (with equal probability) a cluster from the RCL. The second strategy uses the posterior probability matrix produced by EM as a guidance. RCL includes all feasible clusters, but the cluster j is chosen at random according to the probability $P(x|\mu_j, \Sigma_j)$. Intuitively, this means that nodes near the centers of Gaussian mixtures have lower probabilities of getting assigned to a cluster that is very far away. Experiments comparing the second strategy with with the standard RCL strategy with $l = 2, 3, 4$ all heavily favor the utilization of posterior probability. As a result we chose the randomization using the posterior probability in our implementation.

When determining the cluster medians, we construct the RCL by including two nodes that are closest to each mixture means, and select each median randomly from two nodes in the RCL. In our GRASP implementation, we let the above randomization process generate 20 solutions from each EM output of P and μ.

In addition, we use information from previous EM iterations as an adaptive intensification strategy. The approach is inspired by Fleurent and Glover (1999), in which the authors noted that

the basic GRASP disregards information gathered in previous iterations and proposed to use long term memory in GRASP. Similar strategy can also be found in the GRAMPS (Ahmadi and Osman, 2005). The goal is to guide the EM iterations to start from neighborhoods of more promising solutions while also incorporating diversification provided by a random initialization. In our algorithm, the adaptive long-term memory keeps track of the mixture centers from the EM models with highest likelihood. We keep a subset (50%) of the μ_js while allowing the rest to be randomly generated. Algorithm 2 summarizes our implementation of GRASP with EM.

We test the performance of the GRASP algorithm by comparing it with the Algorithm 1 presented earlier. We benchmark the best solution found within a time limit of 15 seconds on test instance p11-p20. Table 5 reports the results. For all but one instances GRASP is able to find a better or equally good solution as a multi-start algorithm, which confirms our implementation of GRASP as an effective diversification strategy. Figure 6 shows the distribution of the solutions found with in 15 seconds. On average the GRASP solutions have higher quality and are less likely to generate unpromising solutions.

Table 5: Best solution found in 15 seconds

	Best Optimality Gap (%)		Number of Solutions Generated	
	GRASP	Multi-Start	GRASP	Multi-Start
p11	0.30	0.50	70	13
p12	0.00	1.45	70	14
p13	0.00	0.00	60	28
p14	1.22	0.41	70	8
p15	1.74	2.02	66	12
p16	0.52	1.78	60	7
p17	0.00	0.00	67	10
p18	1.05	4.51	70	12
p19	1.75	3.30	63	11
p20	1.69	1.89	70	8

In addition to the test instances available in OR-Library, we also use the GRASP procedure to

Algorithm 2 A GRASP for Capacitated EM

$Best\_Likelihood \leftarrow$ MIN_VALUE
$Best\_EM\_\mu \leftarrow$ NULL

while time limit not exceeded **do**
  Step 1. Adaptive EM
  if $Best\_EM\_\mu$ is NULL **then**
    Initialize EM from a random p matrix.
  else
    Initialize EM with half of μ from $Best\_EM\_\mu$, and half from randomly chosen nodes.
  end if
  while not convergence **do**
    M-Step,
    Constrained E-Step.
  end while
  if $Log\_Likelihood > Best\_Likelihood$ **then**
    $Best\_EM\_\mu \leftarrow \mu$
  end if

  for all $iteration = 1, ..., 20$ **do**
    Step 2. Generate Solutions Using Randomized Greedy Procedure
    for all cluster j **do**
      Add closest 2 nodes from μ to RCL,
      Assign a random nodes in RCL as median for cluster j.
    end for

    for all point i **do**
      Let $R_i \leftarrow p_{(1)}^{(i)} - p_{(2)}^{(i)}$,
      where $p_{(1)}^{(i)}, p_{(2)}^{(i)}$ are the largest and second largest entries in row i of P.
    end for
    for all decreasing $i \in R_i$ **do**
      Assign point i to cluster j at random according to probability distribution of p_{ij}.
    end for
    Step 3. Local Search (Omitted)
  end for
end while

Figure 6: Distribution of Solutions Generated

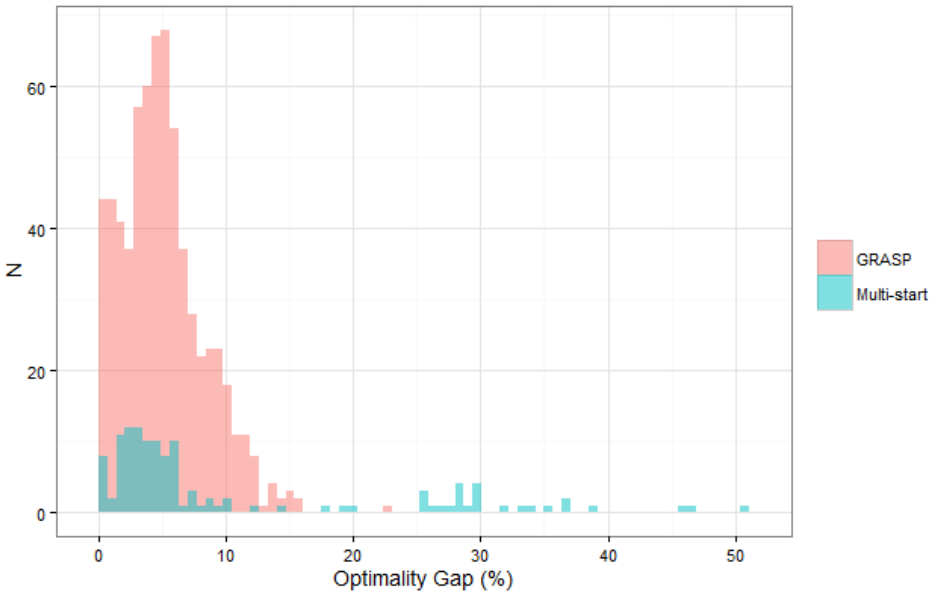
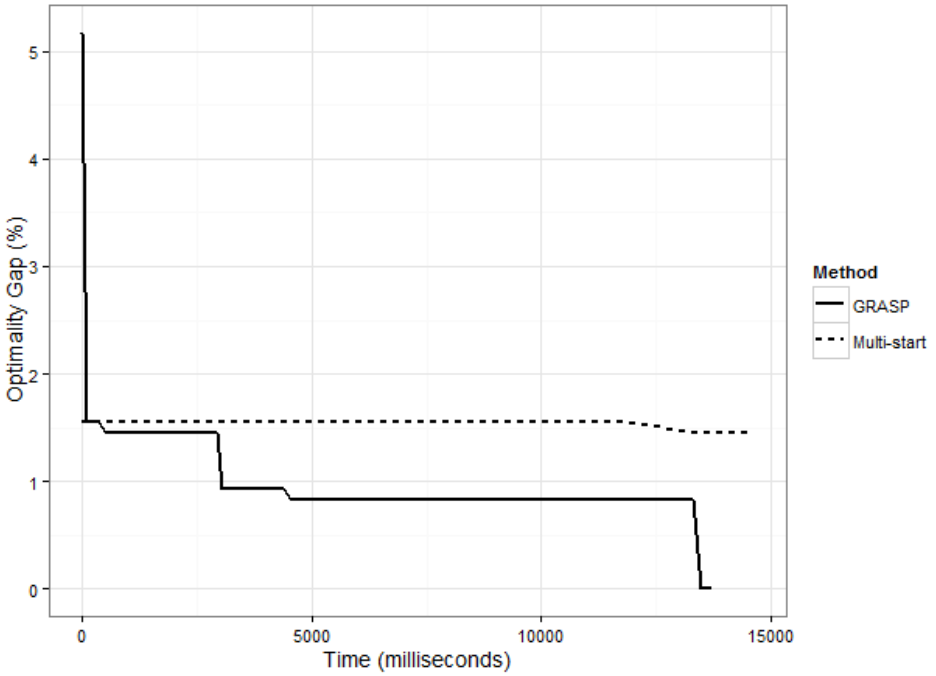


Figure 7: Best solution generated v.s. search time



benchmark the performance on larger instances. The set of six instances corresponds to real data from the Brazilian city of Sao Jose dos Campos. Their dimensions (n, p) are (100,10), (200,15), (300,25), (300,30), (402,30) and (402,40), respectively. However, we notice that the performance of the GRASP is not as good as the performance reported by Chaves and Lorena (2010) using a cluster search algorithm. This is partially due the hybrid meta-heuristics they employ allows more neighborhoods to be searched.

Table 6: Best solution found in 15 seconds

	Best Optimality Gap (%)		Number of Solutions Generated	
	GRASP	Multi-Start	GRASP	Multi-Start
sjc1	1.24	7.12	51	16
sjc2	1.26	1.58	12	5
sjc3a	3.27	5.62	2	2
sjc3b	2.39	3.12	7	1
sjc4a	6.21	6.60	3	1
sjc4b	3.98	4.07	3	1

3.7 Conclusion and Future Research

In this essay we present a new construction heuristic based on the EM algorithm for solving the CCP. A comparison with existing methods was performed to demonstrate the effectiveness of our algorithm. In addition, we conduct analysis on the effect of point pattern distribution of nodes, and extend the algorithm to stochastic variants of CCP.

The challenge of using our approach is that the EM algorithm only guarantees convergence to local optima. This challenge is also faced by many traditional clustering algorithms such as k-means, and the usual procedure to alleviate the problem is using randomized initialization. Besides using a randomized initialization, we proposed a GRASP version of the algorithm for solving larger instances. The results are not on par with more established meta-heuristics such as VNS and

cluster search. This can be partially attributed to the relative simple local search we implemented. Tu et al. (2008) and Jank (2006) proposed a stochastic variation of the EM algorithm, based on genetic algorithm, to search for the global solution and it might provide another promising strategy for the problem.

Our findings can motivate several directions for future research. Model-based clustering is applicable to a wider range of the problems, for example social network of actors (Handcock et al., 2007), genetic data, etc., compared with model-free methods. Through the combination of the OR perspective (i.e. resource constraints), and model-based clustering, the model-based CCP can be extended to answer more interesting problems. One benefit of model-based clustering is that it also provides an approach of choosing the number of clusters using model selection techniques in statistics. This may be used to extend our method to solve a more general problem, e.g., simultaneously deciding the location and the number of service depot for customers, while each service depot has a capacity constraint. Last but not the least, EM algorithm can be adapted to the Map-Reduce computing paradigm (Chu et al., 2007), and it is interesting to investigate how the proposed algorithm can speed up in a multi-core machine or a computer cluster.

Bibliography

Samad Ahmadi and Ibrahim H Osman. Density based problem space search for the capacitated clustering p-median problem. *Annals of Operations Research*, 131(1-4):21–43, 2004.

Samad Ahmadi and Ibrahim H Osman. Greedy random adaptive memory programming search for the capacitated clustering problem. *European Journal of Operational Research*, 162(1):30–44, 2005.

Roberto Baldacci, Eleni Hadjiconstantinou, Vittorio Maniezzo, and Aristide Mingozzi. A new method for solving capacitated location problems based on a set partitioning approach. *Computers & Operations Research*, 29(4):365–386, 2002.

Jeffrey D Banfield and Adrian E Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, pages 803–821, 1993.

Sergio Barreto, Carlos Ferreira, Jose Paixao, and Beatriz Sousa Santos. Using clustering analysis

in a capacitated location-routing problem. *European Journal of Operational Research*, 179(3): 968–977, 2007. ISSN 0377-2217.

Julian Besag and Peter J Diggle. Simple monte carlo tests for spatial pattern. *Applied statistics*, pages 327–333, 1977.

Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Choosing starting values for the em algorithm for getting the highest likelihood in multivariate gaussian mixture models. *Computational Statistics & Data Analysis*, 41(3):561–575, 2003.

Hans H Bock. Probabilistic models in cluster analysis. *Computational Statistics & Data Analysis*, 23(1):5–28, 1996.

Barry N Boots and Arthur Getis. Point pattern analysis. 1985.

Burcin Bozkaya, Erhan Erkut, and Gilbert Laporte. A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*, 144(1):12–26, 2003.

Gilles Celeux and Gérard Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793, 1995.

Abraham Charnes and William W Cooper. Chance-constrained programming. *Management Science*, 6(1):73–79, 1959.

Antonio Augusto Chaves and Luiz Antonio Nogueira Lorena. Clustering search algorithm for the capacitated centered clustering problem. *Computers & Operations Research*, 37(3):552–558, 2010.

Chun-An Chou, Wanpracha Art Chaovalitwongse, Tanya Y Berger-Wolf, Bhaskar DasGupta, and Mary V Ashley. Capacitated clustering problem in computational biology: Combinatorial and statistical approach for sibling reconstruction. *Computers & Operations Research*, 39(3):609–619, 2012.

Cheng Chu, Sang Kyun Kim, Yi-An Lin, YuanYuan Yu, Gary Bradski, Andrew Y Ng, and Kunle Olukotun. Map-reduce for machine learning on multicore. *Advances in neural information processing systems*, 19:281, 2007.

Philip J Clark and Francis C Evans. Distance to nearest neighbor as a measure of spatial relationships in populations. *Ecology*, pages 445–453, 1954.

Elon Santos Correa, Maria Teresinha A Steiner, Alex A Freitas, and Celso Carnieri. A genetic algorithm for solving a capacitated p-median problem. *Numerical Algorithms*, 35(2-4):373–388, 2004.

Abhijit Dasgupta and Adrian E Raftery. Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association*, 93(441):294–302, 1998.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

Juan A Díaz and Elena Fernandez. Hybrid scatter search and path relinking for the capacitated p-median problem. *European Journal of Operational Research*, 169(2):570–585, 2006.

Peter J Diggle, Julian Besag, and J Timothy Gleaves. Statistical analysis of spatial point patterns by means of distance methods. *Biometrics*, pages 659–667, 1976.

Thomas A Feo and Mauricio GC Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations research letters*, 8(2):67–71, 1989.

Krzysztof Fleszar and Khalil S Hindi. An effective vns for the capacitated p-median problem.

European Journal of Operational Research, 191(3):612–622, 2008.

Charles Fleurent and Fred Glover. Improved constructive multistart strategies for the quadratic assignment problem using adaptive memory. *INFORMS Journal on Computing*, 11(2):198–204, 1999.

Stewart Fotheringham and Peter Rogerson. *Spatial analysis and GIS*. CRC Press, 2013.

Chris Fraley and Adrian E Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, 97(458):611–631, 2002.

Kuzman Ganchev, Joao Graça, Jennifer Gillenwater, and Ben Taskar. Posterior regularization for structured latent variable models. *The Journal of Machine Learning Research*, 99:2001–2049, 2010.

Michel Gendreau, Gilbert Laporte, and René Séguin. Stochastic vehicle routing. *European Journal of Operational Research*, 88(1):3–12, 1996.

Fred Glover and Gary A Kochenberger. *Handbook of Metaheuristics*. Springer, 2003.

Mark S Handcock, Adrian E Raftery, and Jeremy M Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 170(2):301–354, 2007.

John C Hudson and Phillip M Fowler. *The Concept of Pattern in Geography, By John C. Hudson and Phillip M. Fowler*. University of Iowa, Department of Geography, 1966.

Wolfgang Jank. The em algorithm, its randomized implementation and global optimization: Some challenges and opportunities for operations research. In *Perspectives in operations research*, pages 367–392. Springer, 2006.

Dimitris Karlis and Evdokia Xekalaki. Choosing initial values for the em algorithm for finite mixtures. *Computational Statistics & Data Analysis*, 41(3):577–590, 2003.

CKY Lin. Stochastic single-source capacitated facility location model with service level requirements. *International Journal of Production Economics*, 117(2):439–451, 2009.

John M Mulvey and Michael P Beck. Solving capacitated clustering problems. *European Journal of Operational Research*, 18(3):339–348, 1984.

Ibrahim H Osman and Nicos Christofides. Capacitated clustering problems by hybrid simulated annealing and tabu search. *International Transactions in Operational Research*, 1(3):317–336, 1994.

IH Osman and S Ahmadi. Guided construction search metaheuristics for the capacitated p-median problem with single source constraint. *Journal of the Operational Research Society*, 58(1): 100–114, 2007.

Soo Byong Park. Performance of successively complex rules for locational decision-making. *Annals of Operations Research*, 18(1):323–343, 1989.

Hasan Pirkul. Efficient algorithms for the capacitated concentrator location problem. *Computers & Operations Research*, 14(3):197–208, 1987.

Adrian E Raftery and Nema Dean. Variable selection for model-based clustering. *Journal of the American Statistical Association*, 101(473):168–178, 2006.

Stephan Scheuerer and Rolf Wendolsky. A scatter search heuristic for the capacitated clustering problem. *European Journal of Operational Research*, 169(2):533–547, 2006.

Yufeng Tu, Michael O Ball, and Wolfgang S Jank. Estimating flight departure delay distributions - a statistical approach with long-term trend and short-term pattern. *Journal of the American Statistical Association*, 103(481):112–125, 2008.

Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geof-

frey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

Ka Yee Yeung, Chris Fraley, Alejandro Murua, Adrian E Raftery, and Walter L Ruzzo. Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17(10): 977–987, 2001.