POMS PRODUCTION AND OPERATIONS MANAGEMENT

PRODUCTION AND OPERATIONS MANAGEMENT POMS
Vol. 27, No. 10, October 2018, pp. 1775–1794
ISSN 1059-1478 | EISSN 1937-5956 | 18 | 2710 | 1775
DOI 10.1111/poms.12737

Parallel Aspect-Oriented Sentiment Analysis for Sales Forecasting with Big Data

Raymond Yiu Keung Lau
Department of Information Systems, College of Business, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong, raylau@cityu.edu.hk

Wenping Zhang, Wei Xu
School of Information, Renmin University of China, Beijing, 100872, China, wpzhang@ruc.edu.cn, weixu@ruc.edu.cn

While much research work has been devoted to supply chain management and demand forecast, research on designing big data analytics methodologies to enhance sales forecasting is seldom reported in existing literature. The big data of consumer-contributed product comments on online social media provide management with unprecedented opportunities to leverage collective consumer intelligence for enhancing supply chain management in general and sales forecasting in particular. The main contributions of our work presented in this study are as follows: (1) the design of a novel big data analytics methodology that is underpinned by a parallel aspect-oriented sentiment analysis algorithm for mining consumer intelligence from a huge number of online product comments; (2) the design and the large-scale empirical test of a sentiment enhanced sales forecasting method that is empowered by a parallel co-evolutionary extreme learning machine. Based on real-world big datasets, our experimental results confirm that consumer sentiments mined from big data can improve the accuracy of sales forecasting across predictive models and datasets. The managerial implication of our work is that firms can apply the proposed big data analytics methodology to enhance sales forecasting performance. Thereby, the problem of under/over-stocking is alleviated and customer satisfaction is improved.

Key words: big data analytics; parallel sentiment analysis; machine learning; sales forecasting
History: Received: January 2016; Accepted: June 2017 by Ram Ganeshan, after 2 revisions.

1. Introduction

Big data holds the promise to provide firms with competitive business intelligence to enhance their strategic decision-making processes (Lau et al. 2016, McAfee and Brynjolfsson 2012). However, while much research work has been devoted to supply chain management and demand forecast (Boyaci and Gallego 2004, Breiter and Huchzermeier 2015, David and Adida 2015, Sun et al. 2008, Tang et al. 2014, de Treville et al. 2014), research about the design of big data analytics methodologies for enhancing sales forecasting is seldom reported in existing literature. Recent studies have demonstrated the potentials of mining consumers' sentiments embedded in online communications for enhancing the performance of sales forecasting (Ahn and Spangler 2014, Archak et al. 2011, Rui et al. 2013, Sonnier et al. 2011, Yu et al. 2012). Nevertheless, these studies have not examined the problem of mining sentiments from a huge volume of consumer-contributed product comments (i.e., big data) for sales forecasting. Our work reported in this study aims to fill the aforementioned research gaps.

In the era of big data, it is believed that the sheer volume of user-contributed comments on online social media can be leveraged by firms to enhance various applications such as stock investment, business merger and acquisition, customer relationship management, and so on (Bollen et al. 2011, Lau et al. 2012, McAfee and Brynjolfsson 2012). However, traditional computational methods simply cannot scale up with big data (Canny and Zhao 2013, Gandomi and Haider 2015). Indeed, it is well acknowledged that designing big data analytics methodologies that can effectively and efficiently process big data is a very challenging research problem (Goes 2014, Lau et al. 2016). The main contributions of our research work are twofold. First, we design a novel big data analytics methodology that is underpinned by a parallel aspect-oriented sentiment analysis algorithm for efficient mining of consumer sentiments from big data. Second, we design and empirically evaluate a sentiment enhanced sales forecasting method that is empowered by a parallel co-evolutionary extreme learning machine. To our best knowledge, this is the first successful research on parallel mining of aspect-oriented sentiments from big data for enhancing sales forecasting performance.

In general, sentiment analysis concerns about analyzing people's attitudes (e.g., positive or negative) towards entities (e.g., products or services) (Liu 2012, p. 1). More specifically, aspect-oriented sentiment analysis is the most fine-grained form of sentiment analysis in which the opinion targets are the attributes (e.g., screen) of an entity (e.g., camera), and such a sentiment analysis approach is believed to be more applicable to support real-world applications (Liu 2015, p. 90). By mining aspect-oriented sentiments from a huge volume of consumer-contributed product comments, firms can develop deeper insights about the specific aspects of products that consumers like or do not like. Accordingly, firms can leverage such a consumer intelligence to enhance sales forecasting, marketing, product design, and other business functions. Traditional sales forecasting methods often adopt a univariate time series approach in which predictions about future sales mainly rely on historical sales data (Weiss 1984, Wong and Guo 2010). However, future sales are likely to be influenced by volatile consumer demands as well. This is particularly true for the consumer electronic products that are traded at giant electronic commerce platforms such as Amazon.com, JD.com, Taobao.com, and so on. The basic intuition of the proposed sentiment-based sales forecasting method is that we combine both historical sales patterns and consumers' latest preferences extracted from big data to bootstrap the performance of sales forecasting.

The managerial implications of our research can lead to more effective sales forecasting, thereby enabling firms to improve their supply chain management, just-in-time (JIT) delivery, corporate financial planning, marketing, and customer relationship management. The rest of the article is organized as follows. The next section discusses related work about big data analytics, aspect-oriented sentiment analysis, and sentiment-based forecasting. An overview of the proposed big data analytics methodology, and the illustration of the related computational methods are given in section 3. Section 4 reports the experimental results of the proposed parallel aspect-oriented sentiment analysis method and that of the sentiment enhanced sales forecasting method. Finally, we offer concluding remarks and discuss future directions of our research work.

2. Related Work

2.1. Big Data Analytics

Big data is characterized by the five Vs: volume, velocity, variety, veracity, and value (Lau et al. 2016). Among the leading big data research groups, Canny and Zhao (2013) developed the BID big data analytics suite for supporting parallel machine learning computations at UC Berkeley. Doulkeridis and Nørvåg (2014) discussed the potentials and limitations of the well-known MapReduce parallel processing framework for large-scale query processing. Motivated by the pioneer work of Jindal and Liu (2008), Zhang et al. (2014) developed a parallel co-evolutionary genetic algorithm for detecting deceptive comments on online social media, and hence to address the veracity issue of big data. Ramirez-Gallego et al. (2015) have reported their big data analytics solution developed based on the in-memory parallel processing framework named Spark. Gandomi and Haider (2015) pinpoint future research directions of designing big data analytics solutions to address issues related to the 5Vs of big data.

By using a cluster of computing nodes, Newman et al. (2007) developed the Approximate Distributed Latent Dirichlet Allocation (AD-LDA) model for large-scale text analytics. In AD-LDA, the Gibbs sampling process was conducted on each processor independently, and a global update of the sampling results was performed after a number of iterations. Due to the lack of coordination among distributed Gibbs sampling, serious biases of topic distributions might be generated from the AD-LDA model. To alleviate such a problem, Wang et al. (2009) developed the Parallel Latent Dirichlet Allocation (PLDA) model where a global word-topic matrix was maintained, and each processor of a cluster updated the global word-topic matrix periodically. However, dead locks of concurrent updating and excessive communication overheads would occur when many local processors tried to update the global matrix at the same time. By introducing auxiliary variables into the Hierarchical Dirichlet Process (HDP), Williamson et al. (2013) developed the distributed HDP where a big dataset was first divided into data points according to the auxiliary variables which captured the statuses of distributed processors. Then, distributed Dirichlet processes toward these data points were conducted conditioned on the allocated processors. Nevertheless, the computational efficiency of the distributed HDP model would be seriously affected by imbalanced workload distribution along a cluster.

Though our research also aims to address the issues of volume and velocity related to big data, it differs from all the aforementioned studies in that we exploit a massively parallel computational approach to scale up sentiment analysis and sales forecasting for big data. Compared to the aforementioned parallel topic modeling methods, the proposed topic model for parallel aspect extraction extends the classical HDP model such that there is not a compulsory data split conditioned on allocated processors. Consequently, it is possible to distribute balanced workload along a cluster.

2.2. Aspect-oriented Sentiment Analysis

Sentiment analysis aims to uncover people's attitudes (e.g., positive or negative) toward entities (e.g., products), and it can be conducted at document, sentence, or aspect levels (Liu 2012, p. 4). Aspect-oriented sentiment analysis is most applicable to real-world applications, and it involves aspect extraction and aspect sentiment classification (Liu 2015, pp. 90–91). The seminal study from Hu and Liu (2004) is the first to discuss the problem of aspect (also called feature) extraction from product reviews. In particular, they applied the Apriori association rule mining algorithm to extract explicit product features (e.g., noun phrases) frequently occurring in product reviews. In general, the main approaches for aspect sentiment classification include lexicon-based method (Ding et al. 2008), linguistic rule-based method (Ding and Liu 2007), graph-based method (Wu et al. 2011), and supervised learning method (Yang and Cardie 2013). Given the simplicity of lexicon-based method (Ding et al. 2008), it has the potential to be applied to big data. However, an empirical evaluation of such an approach under a big data context is needed.

Mei et al. (2007) developed the topic-sentiment mixture (TSM) model that could simultaneously extract dynamic topics (i.e., aspects) and sentiments from weblogs. Titov and McDonald (2008) proposed the supervised Multi-aspect Sentiment (MAS) model that utilized user-rated aspects to guide the extraction of topics corresponding to some product aspects. Furthermore, some researchers extended the classical Latent Dirichlet Allocation (LDA) model to mine product aspects based on seeding sentiment indicators (Xu et al. 2012, Zhao et al. 2010). Chen et al. (2013) developed the LDA with m-set and c-set (MC-LDA) topic model that utilized the must-link set and the cannot-link set as prior knowledge to guide multi-sense aspect extraction. In a similar vein, Chen et al. (2014) proposed the Automated Knowledge LDA (AKL) topic model that could leverage a cluster of topics extracted from multiple domains for fault-tolerant aspect extraction. Moreover, lifelong learning and the big data of product reviews were applied to automatically acquire prior domain knowledge represented as must-links and cannot-links to guide aspect extraction in the AMC topic model (Chen and Liu 2014). In contrast, Lau et al. (2014) applied LDA and probabilistic language modeling methods to extract a hierarchy of product aspects without using prior knowledge. Wang et al. (2014) developed the Fine-grained Labeled LDA (FL-LDA) and Unified Fine-grained Labeled LDA (UFL-LDA) models for mining meaningful aspects by using seeding aspects extracted from e-Commerce Websites. Recently, Wang et al. (2016) have developed the Lifelong Aspect-based Sentiment Topic (LAST) model that simultaneously handled aspect extraction, opinion identification, polarity classification, and separation of general and aspect-specific opinions according to the generalized Polya Urn (GPU) principle.

Though topic modeling methods have been widely used for aspect extraction, these methods are computationally very expensive. In fact, topic modeling methods are often applied to benchmarking exercises that evaluate the efficiency of big data analytics solutions (Canny and Zhao 2013). Our work differs from the aforementioned studies in that we design a novel aspect-oriented sentiment analysis method that is underpinned by a parallel Gamma–Poisson-enhanced Hierarchical Dirichlet Process (GP-HDP) topic model to alleviate the computational bottleneck of topic-based aspect extraction under the environment of big data.

2.3. Sentiment-based Predictions

Traditional time series-based sales forecasting methods often adopt the Box-Jenkins Autoregressive Integrated Moving Average (ARIMA) model (Gep and Da 1970) or its variants which are based on the assumption that the generation process of a time series is relatively stable (Lu et al. 2012, Sun et al. 2008, Wong and Guo 2010). Unfortunately, such an assumption may not hold for many real-world sales forecasting domains. Recently, researchers have tried to exploit consumers' demands implicitly captured in online product comments for enhancing sales forecasting performance (Ahn and Spangler 2014, Archak et al. 2011, Chevalier and Mayzlin 2006, Sonnier et al. 2011, Yu et al. 2012). The seminal paper from Chevalier and Mayzlin (2006) examined the impact of online reviews on the sales of books at two Websites. Archak et al. (2011) applied machine learning techniques to extract product feature-based sentiments to predict the sales ranks (i.e., the proxy of actual sales) of two product categories (e.g., digital cameras and camcorders) at Amazon. By using a proprietary sales dataset, Sonnier et al. (2011) developed a dynamic structured model that utilized sentiments embedded in consumers' online communications for predicting future sales. The autoregressive sentiment-aware (ASRA) model (Liu et al. 2007, Yu et al. 2012) and autoregressive sentiment and quality aware (ARSQA) model (Yu et al. 2012) utilized probabilistic latent semantic analysis technique to extract implicit sentiments from online movie reviews for predicting movie sales. Rui et al. (2013) also found that positive Twitter sentiments were associated with larger movie sales. Ahn and Spangler (2014) incorporated both search volumes and sentiments of products into the ARIMA model for enhancing monthly sales forecasting of cars. In fact, user sentiments were also found useful for enhancing time series prediction for stock (Bollen et al. 2011, Das and Chen 2007, Si et al. 2014).

Our work differs from the aforementioned sentiment-based forecasting methods in two ways: (1) we applied an aspect-oriented sentiment analysis method to mine context-sensitive sentiments from product comments to enhance sales forecasting performance, whereas previous studies often adopted a context-free sentiment analysis approach; (2) we aim to alleviate the computational bottleneck of topic modeling-based aspect extraction from big data by exploiting a novel parallel topic modeling method. We believe that aspect-oriented sentiment analysis is more appropriate for sentiment-based sales forecasting because it can more accurately capture consumers' real product preferences. For instance, the sentiment word "small" is assumed to carry a negative polarity in well-known sentiment lexicons such as SentiWordNet (Esuli and Sebastiani 2005) and SenticNet-3 (Cambria et al. 2014). However, this sentiment word may actually carry a positive polarity in consumer electronics domains such as "my small size notebook is ideal for traveling". In this case, the sentiment "small" for the aspect "size of a notebook" (i.e., the context) has a positive polarity. In contrast, since existing sentiment lexicons usually do not consider the polarity of a sentiment with respect to a specific context such as a product domain or a product aspect (i.e., context-free), it is possible to make mistakes in sentiment analysis. Accordingly, it affects the accuracy of subsequent sales forecasting. Moreover, sentiment words (e.g., "good") alone may not express any consumer sentiment at all in a product review (e.g., "Is the Samsung Galaxy Note 7 good?"). So, extracting sentiments with respect to specific product aspects is a better way to uncover consumers' real preferences.

3. A Big Data Analytics Methodology for Sales Forecasting

3.1. System Architecture

The system architecture of the Parallel Aspect-oriented Sentiment Analysis for Demand forecasting (PASAD) service built on top of Apache Spark[1] is depicted in Figure 1. Due to limited space, we will only illustrate the computational details of the parallel product ontology miner, parallel sentiment analyzer, and product demand estimator in this study. Some sample program source codes and review data are made available via our project Website.[2] The proposed service consists of six layers as described below:

(1) Storage Layer: the storage layer leverages Apache HDFS for permanent storage and retrieval of big data (e.g., the sheer volume of consumer-contributed product comments) that are collected by our dedicated crawlers and external APIs.

(2) Batch Layer: the batch layer utilizes the MapReduce model of Apache Hadoop[3] for processing big data in batch mode. Apache's Yet Another Resource Negotiator (YARN) and Mesos are distributed operating systems that seamlessly connect Spark and Hadoop running on a cluster of commodity computers.

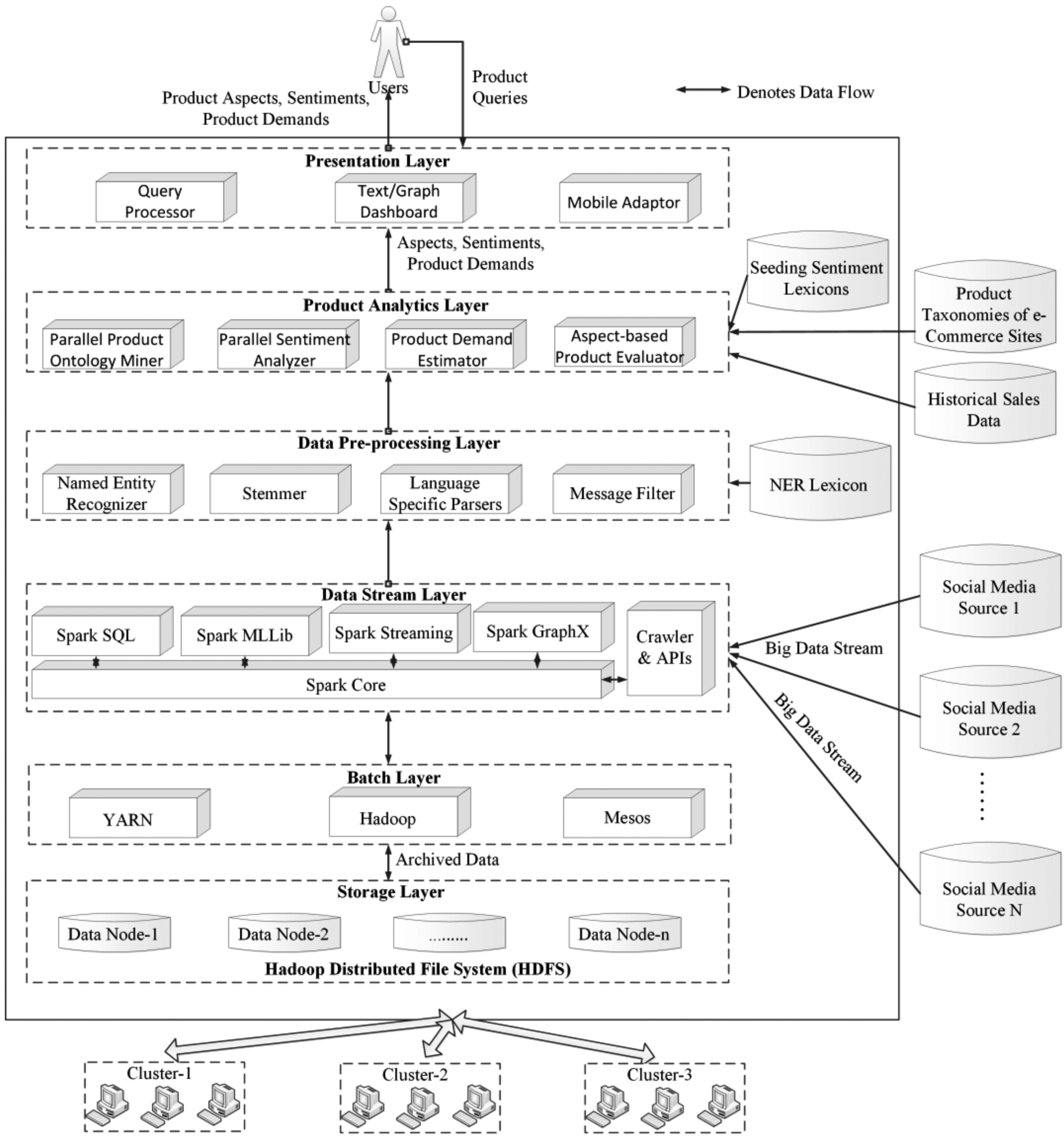
(3) Data Stream Layer: the data stream layer leverages the Spark streaming module, and dedicated crawlers and APIs (e.g., the Datasift Twitter firehose and the Sina Weibo crawler) for retrieving a big data stream of product comments and product related messages from external data sources.

(4) Data Pre-processing Layer: the data pre-processing layer invokes a message filter to filter out non-relevant user-contributed comments. Moreover, the language-specific parsers are applied to process incoming messages written in different languages. Our current implementation includes a Chinese word segmentation and part-of-speech (POS) tagging module named ICTCLAS[4] developed by the Chinese Academy of Sciences. For English messages, the stemmer module converts all English words to their root forms. For user comments retrieved from a general purpose social media source (e.g., Twitter or Sina Weibo), an extended Named Entity Recognition (NER) module developed based on GATE (Maynard et al. 2001) is applied to identify which product(s) a user comment refers to.

(5) Product Analytics Layer: the product analytics layer consists of a *parallel product ontology miner* that can alleviate the computational bottleneck of topic modeling-based ontology learning from big data. In addition, the *parallel sentiment analyzer* extracts and classifies aspect-oriented sentiments from a big data stream of product comments by leveraging the Spark streaming module and the periodically refined product ontology. The *product demand estimator* makes use of aspect-oriented consumer sentiments and historical sales data to predict future sales of a specific geographical area at a pre-defined time point. Finally, the *aspect-based product evaluator* pinpoints the major weaknesses or strengthens of a product based on aspect-oriented consumer sentiments.

(6) Presentation Layer: the presentation layer conveys aspect-oriented sentiments and sales forecasting results to users by using the analytics dashboard. Moreover, the mobile adaptor can adjust presentation formats and layouts of the system outputs for proper rendering on different mobile devices. The query

Figure 1 The System Architecture of PASAD



processor accepts users' queries about certain product or product categories of a geographical area. Figure 2 is a snapshot view of the analytics dashboard of the proposed PASAD service. The "Location" field depicts the specific geographical area where the review writers are located. The "Forecast" field shows the predicted sales volume of a specific period (e.g., the following day or week) as specified by a user query. The original reviews and the consumer sentiment summary charts by day, week, and month are shown on the dashboard.

3.2. Parallel Product Ontology Miner

The automatically constructed product ontology can be seen as an advanced form of sentiment lexicon. Not only does it capture aspect-oriented

Figure 2 A Snapshot View of PASAD [Color figure can be viewed at wileyonlinelibrary.com]

sentiments but it also contains a product taxonomy. Ontology is generally considered as a formal specification of conceptualization which consists of concepts and their relationships (Gruber 1993). More specifically, the proposed product ontology leverages fuzzy sets and fuzzy relations (Zadeh 1965) to capture the uncertainty arising in product aspect extraction and opinion identification. The formal definition of the proposed product ontology is given below.

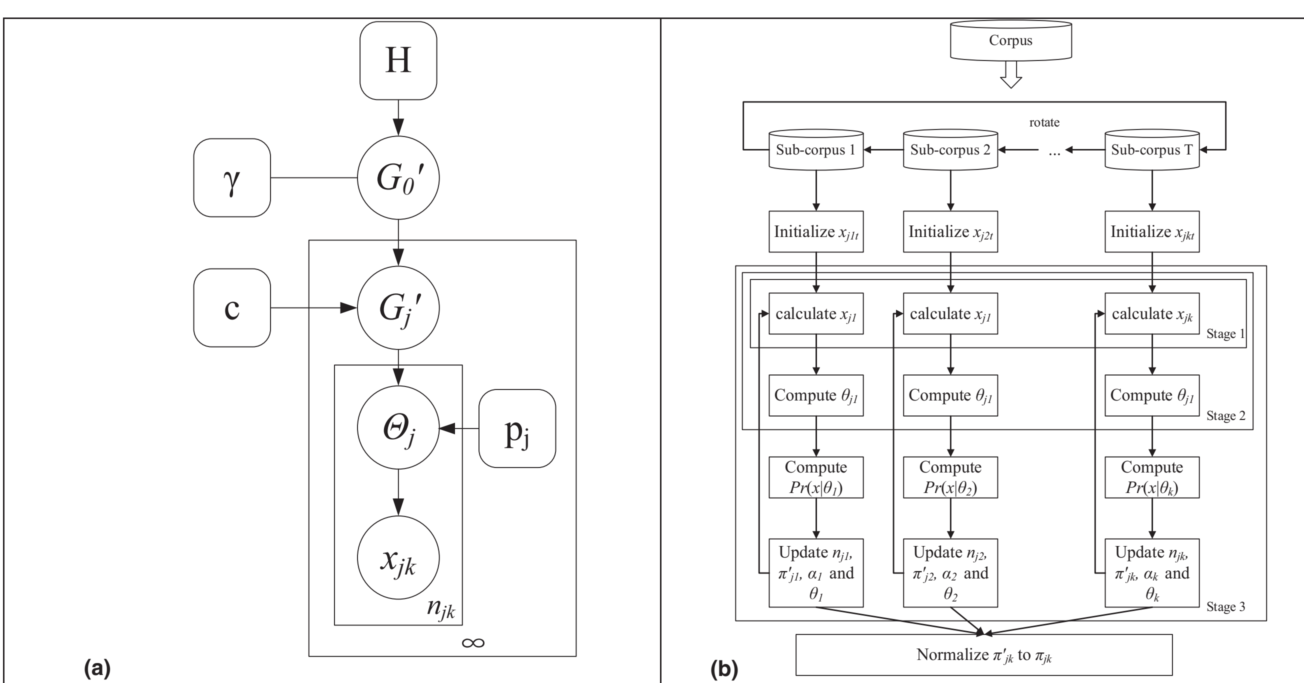
Definition 1. *Product Ontology: A domain-specific product ontology is a nonuple* $Ont := \langle P, A, S, PO, R_{P\times P}, R_{A\times A}, R_{A\times S\times PO}, U, T\rangle$, *where* P, A, S *refer to finite sets of products, aspects, and sentiments, respectively. The polarity set* $PO = \{pos, neg, neu\}$ *captures the orientations of sentiments. The product taxonomy relation* $R_{P\times P}$ *is characterized by the fuzzy membership function* $u_{R_{P\times P}} : P \times P \mapsto [0, 1]$ *which represents the strength of the sub-class/super-class relations among a set of products* P. *The product aspect taxonomy relation* $R_{A\times A}$ *is characterized by the fuzzy membership function* $u_{R_{A\times A}} : A \times A \mapsto [0, 1]$. *The aspect-sentiment-polarity association is defined by:* $R_{A\times S\times PO} : A \times S \times PO \mapsto [-1, +1]$, *where the range of this relation represents the polarity strength of an aspect-sentiment pair* (a_i, s_i). *Since the product ontology is mined based on the consumer-contributed product comments,* U *refers to the set of consumers who adopt the current ontological view at the set of time points* T.

Since the descriptions of products P and the corresponding product taxonomy $R_{P\times P}$ are easily retrieved from e-Commerce sites (e.g., Amazon, Newegg, JD.com, Dianping, etc.), we can apply ontology mapping techniques (Kalfoglou and Schorlemmer 2003) to consolidate the final product taxonomy of a chosen language. Accordingly, this study focuses on learning the aspect taxonomy and aspect-oriented sentiments from consumer-contributed product comments. In particular, the proposed method automatically learns the aspect taxonomy $R_{A\times A}$ and aspect-sentiment associations $R_{A\times S\times PO}$ for each product category, and then these ontology segments are combined to form the final product ontology. Though recent studies show that topic modeling methods are effective to mine aspect taxonomies (Lau et al. 2014, Wang et al. 2014, Xu et al. 2012, Zhao et al. 2010), topic modeling is computationally very expensive (Canny and Zhao 2013). Accordingly, one main contribution of our research is to develop a novel parallel topic model to alleviate such a computational bottleneck. On the other hand, despite previous studies often adopted LDA-based aspect extraction methods, the number of topics (i.e., aspect classes) must be manually defined in advance. Practically, it is quite difficult to estimate the optimal number of topics for aspect extraction. To alleviate such a problem, we develop the parallel Gamma–Poisson-enhanced Hierarchical Dirichlet Process (GP-HDP) model for aspect extraction and grouping. The advantage of the proposed model is that the optimal number of topics can be automatically inferred according to the inherent characteristics of a dataset.

3.2.1. Parallel Product Aspect Extraction. Similar to the previous study (Wang et al. 2014), product descriptions and consumer-contributed product comments that contain elementary aspects (i.e., product features) are grouped together such that semantically related aspects and high-level aspects can be discovered via the proposed GP-HDP topic model. Compared with the classical Hierarchical Dirichlet Process (HDP) driven by the Steak Breaking or Chinese Restaurant Franchise (CRF) process (Teh et al. 2006), the distinct advantage of the proposed GP-HDP model is that the topic (i.e., high-level aspect) generation processes are completely independent from each other. Such independence lays the solid foundation of the proposed parallel Gibbs sampling algorithm to be illustrated in this section later on. Since directly inferring the parameters of a topic model (e.g., GP-HDP) is computationally very expensive, approximation method such as Gibbs sampling has been widely used for model learning (Cheng and Liu 2014, Lau et al. 2014). Gibbs sampling is based on the Markov Chain Monte Carlo (MCMC) method in which a Markov chain (i.e., a model of transitions among successive states) is established by repeatedly drawing a latent topic for each observable term, based on its conditional probability over other variables (Geman and Geman 1984). In short, by repeatedly drawing samples from a corpus, a probability distribution (e.g., a topic-word distribution) is established to approximate the true underlying distribution. The common Plate notation is applied to describe the generative processes of the GP-HDP models in Figure 3a.

The Gamma–Gamma–Poisson process (the left-hand side of Figure 3) is described as follows. Given a corpus (e.g., a set of product comments), the global topic space G'_0 is drawn by a Gamma process $G'_0 = \sum_{k=1}^{\infty} \alpha_k \delta_{\Phi k} \sim GaP(\gamma, H)$ with the concentration parameter $\gamma \in [0, 1]$ and a prior distribution H which can be drawn from a Dirichlet distribution. The term α_k represents the weight for the kth topic. For a document (e.g., a product comment), draw group topic space $G'_j = \sum_{k=1}^{\infty} \pi'_{jk} \delta_{\Phi k} \sim GaP(c, G'_0)$ with parameter c. π'_{jk} is the weight for the kth topic in the jth document. For each topic (e.g., a high-level aspect), (1) draw p_j from a Beta distribution $p_j \sim Beta(a_0, b_0)$, with parameters a_0 and b_0; (2) establish the document level parameter φ_j from a Gamma distribution $\theta_j \sim GaP(p_j, G'_j)$; (3) draw $X_j = \sum_{k=1}^{\infty} n_{jk} \delta_{\pi k} \sim PoisP(\theta_j)$, where X_j refers to the word set of the document j. n_{jk} represents the number of words that belongs to topic k in document j. Finally, generate the words (e.g., elementary aspects) for each document j according to n_{jk}, and the conditional word distribution given the topic. The

Figure 3 Graphical Model of GP-HDP (Left), and Schemata View of Parallel GP-HDP (Right)



schemata view for the parallel and distributed execution of the GP-HDP model is shown in Figure 3b.

One of the main differences between the classical HDP model and the proposed GP-HDP model lies on the way how these models estimate the variables n_{jk} (i.e., the number of words that belongs to topic k in document j) and N_j (i.e., the total number of words in document j). For the classical HDP model, N_j is a constant (as in typical document processing situations) and it is given to the model. Then, the model invokes the sampling process and updates the count variable n_{jk}. When the count variable n_{jk} for a particular topic $k = k^*$ is updated, all the other count variables $n_{jk}, k \neq k^*$ will be affected because the equality $\sum_{k=1}^{K} n_{jk} = N_j$ must be maintained. This is the very reason why there is a strong topic dependency among the topic sampling processes for the classical HDP model. In contrast, we sample and update n_{jk} first, and then N_j is implicitly determined after a certain number of sampling processes for the proposed GP-HDP model. Please note that the variable n_{jk} is defined for the inner plate (the loop for a document) of the GP-HDP model in Figure 3a (the left-hand side of Figure 3). Essentially, it indicates that the total number of words in document j is implicitly defined based on n_{jk}. The advantage of such a formalization of the GP-HDP model is that sampling and updating of n_{jk} for one topic will not affect the sampling processes of other topics simply because the constraint $\sum_{k=1}^{K} n_{jk} = N_j$ is no longer imposed in this case. Accordingly, the topic (i.e., high-level aspect) generation processes of the proposed GP-HDP model are independent from each other, which lay the foundation for a massively parallel aspect extraction process.

For the classical HDP model, the term N_j is a predefined constant, and n_k is established by sampling. In contrast, for the GP-HDP model, N_j is estimated after the generation of n_{jk}. This may cause a computational problem when the Gamma–Gamma Poisson process is applied to topic modeling in practice. However, by introducing an adaptive parameter m, the Gamma–Gamma–Poisson generative process can be simplified as follows (Liu et al. 2015):

$$\alpha_k|\alpha_0 \sim Gamma\left(\frac{\alpha_0}{K}, 1\right)$$
$$\alpha_{jk}|\alpha_k, p_j \sim Gamma(\alpha_k, 1)$$
$$n_{jk}|m, \pi'_{jk} \sim Pois(m\pi'_{jk})$$
$$\phi_k \sim F$$
$$x_{jk}|\phi_k, n_{jk} \sim p(X|\phi_k, n_{jk}), \tag{1}$$

where F refers to the predefined count distribution for ϕ_k, e.g., a Dirichlet or a Negative Binomial distribution (Zhou and Carin 2012). The joint distribution is then derived as follows.

$$\begin{aligned} p\left(n_{jk}, \pi'_{jk}, x_{jk}, \alpha_k, \Phi_k\right) = {} & \Pi_{k=1}^{K} \frac{\alpha_k^{\frac{\alpha_0}{K}-1}}{\Gamma \frac{\alpha_0}{K}} e^{-\alpha_k} \Pi_{k=1}^{K} \\ & \times \Pi_{j=1}^{J} \frac{\pi_{jk}^{\alpha_{k-1}}}{\Gamma(\alpha_k)} e^{-\pi'_{jk}} \frac{(m\pi'_{jk})^{n_{jk}}}{n_{jk}!} e^{-m\pi'_{jk}} \\ & \times \Pi_{k=1}^{K} F(\Phi_k) \Pi_{k=1}^{K} \Pi_{j=1}^{J} \\ & \times \Pi_{i=1}^{n_{jk}} p(x_{jk}^{(i)}|\Phi_k). \end{aligned} \tag{2}$$

We develop a novel parallel Gibbs sampling algorithm to approximate $\alpha_k, \Phi_k, \pi'_{jk}$, and n_{jk} according to Equations (1) and (2). The sampling and updating procedures for $\alpha_k, \Phi_k, \pi'_{jk}$ are straightforward, and they are derived based on the prior work (Teh et al. 2006, Zhou and Carin 2012). As discussed above, one problem of the Gamma–Gamma–Poisson process is that $N_j (N_j = |X_j|)$ cannot be predefined. Accordingly, we apply the Metropolis–Hasting step based on the following extended Markov Chain Monte Carlo process to deal with such a problem:

To simplify model representation and computational complexity of a sampling process, prior research applied a finite approximation for the number of topics K (Cheng and Liu 2014, Zhou and Carin 2012). However, it is difficult to estimate the upper-bound of K in advance due to the extraordinary high volume of incoming data for big data analytics applications. Consequently, it may end up with insufficient tables (topics), which will introduce serious bias to the sampling process. To address such a problem, we propose a novel strategy of increasing K by Q percent ($k \leftarrow (1 + Q)K$).) Here, Q acts as an adaptive parameter of K. We divide the processors of a cluster into manager processors (P_M) and execution processors (P_E). The manager processors preprocess the data, schedule the sampling tasks, and monitor the progress of the execution processes. Meanwhile, the execution processors manage the threads of independent Gibbs sampling for topics. A P_M processor manages multiple P_E processors, while a P_E processor manages multiple threads (*Thr*). Each thread runs a sampling task for a topic. The tasks are assigned in a top-down manner, while the execution status is reported in a bottom-up direction.

Since workload balancing is a critical issue for parallel Gibbs sampling processing, we advocate the circular queue approach to address such an issue. Suppose we have a dataset j (e.g., a set of product comments). The workload for sampling its sub-elements (e.g., documents) d_i is defined by l_{d_i}. The workload of a sampling task is measured in terms of both document length and the number of unique words in a document. Then, the dataset j is expressed by a vector $D(j) = \langle(d_1, l_{d_1}), (d_2, l_{d_2}), \ldots, (d_n, l_{d_n})\rangle$. We first create a balanced workload circular queue according to the workload of each document. Then, the manager processors P_M are uniformly assigned to the circular queue. In other words, the circular queue is cut into data blocks (DBs) (i.e., RDDs by using Spark's terminology) with a relatively balanced workload. The data points of the circular queue are transferred in a predefined direction (e.g., clockwise). That is, if one data point has been sampled by the execution processors P_E managed by the corresponding manager processor P_M, it will be transferred to the next manager processor. Such a transfer can be done in parallel. Moreover, when an execution processor terminates and transfers control back to its manager processor, the manager processor will communicate with the manager processors preceding and behind it in the circular queue.

The proposed Parallel Gibbs Sampling Algorithm (PGSA) for aspect extraction from a big corpus X is depicted in Figure 4. Given a big dataset X, the PGSA algorithm first assigns subsets of the corpus to the manager processors P_M after initial workload balancing and word bundling. For each manager processor P_M, it allocates tasks to its execution processors P_E with appropriate background information such as documents and number of topics. For each execution processor, it creates a thread to sample each topic according to the updating rules depicted in Table 1. After a certain number of sampling rounds *NIter*, the execution processor reports the status (e.g., word

Figure 4 The PGSA Algorithm for Parallel Aspect Extraction from Big Data

Inputs: Data Corpus X; Pre-estimated max topic number K_0; Adaptive parameter Q; parameter α, m; Number of iterations to update and report $NIter$; Portion of manager processor and execution processor $R, R \in (0,1)$.
Outputs: Number of topics $K_{occupied}$; mixture component ϕ_k and corresponding weight π_{jk}.
Divide processors into P_M and P_E according to R;
Generate balanced workload BWQ_X for Corpus X using the circular queue method;
Uniformly distribute BWQ_X, K_0 and $\{P_E\}$ to manager processors $\{P_M\}$;
for each P_M in $\{P_M\}$
| Arrange $\frac{K_0}{|\{P_M\}|*|\{P_{EM}\}|}$ free topics to each P_E in $\{P_{EM}\}$//$\{P_{EM}\}$ is the set of P_E managed by the current P_M;
| **repeat**
| **for** each P_E managed by P_M
| | Fetch a document j from data blocks on P_M;
| | Generate balanced workload circular queue BWQ_j for j;
| | Initialize the starting point of each thread;
| | Randomly initialize α_k,ϕ_k and π'_{jk};
| | **for** each thread/topic k
| | | **for** $nIter$ from 1 to $NIter$
| | | Update n_{jk} according to the rules depicted in **Table 1**;
| | | Update π'_{jk} according to $Gamma(n_{jk} + \alpha_k, m + 1)$;
| | | **end for**
| | | Update α_k according to $Gamma\left(\frac{\alpha_0}{K} + J, \sum_{j=1}^{J} -log(\pi'_{jk})\right)$;
| | | Update ϕ_k according to $\phi_k \propto F(\phi_k) \prod_{i=1}^{n_i} p(x_k^{(i)}|\phi_k)$;
| | **end for**
| | Report status of current P_E to its manager P_M;
| **end for**
| Send the data block to next P_M when a sampling process finishes;
| Communicate with the P_M preceding and behind it;
| **If** no. of free topic is not sufficient, $K_0 \leftarrow K_0 * (1 + Q)$
| Distribute the newly added topics to each P_E;
| **until Convergence**
end for
Normalize $\{\alpha_k\} \rightarrow \{\pi_{ok}\}$ and $\{\pi'_{jk}\} \rightarrow \{\pi_{jk}\}$;
Return Number of topics $K_{occupied}$; mixture component ϕ_k and corresponding weight π_{jk};

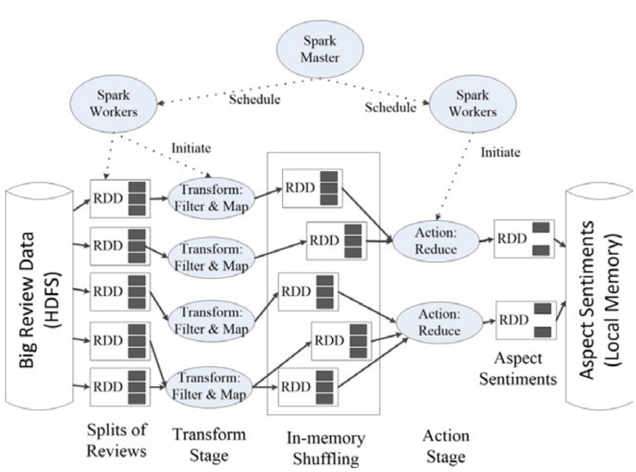
Table 1 Updating Rules for Model Parameter n_{jk}

1. Build a stack S_j to store X'_j for topic k.
2. Draw $u \sim Uniform(0, 1)$
3. If $u < 0.5$, propose an increment of n_{jk} ($n_{jk} \leftarrow n_{jk} + 1$); pop a new word x^* from the top of S_j and add x^* to X'_{jk}. Then, assign it to topic k with the acceptance rate $min\left(1, \frac{m\pi'_{jk}}{n_{jk}+1} px^*(k)\right)$. If it fails, return it to S_j.
4. If $u \geq 0.5$, propose an decrement of n_{jk} ($n_{jk} \leftarrow n_{jk} + 1$); randomly choose and delete a word x^* from X'_{jk} with acceptance rate $min\left(1, \frac{n'_{jk}}{m\pi'_{jk}} \frac{1}{px^*(k)}\right)$. If it fails, return it to S_j.

distribution, number of topics assigned) to its manager. Then, the manager communicates to other managers preceding and behind it in the circular queue. If the manager finds that there are not sufficient free topics, it will increase the pre-estimated topic number K_0 by a rate of Q. These newly added free topics are distributed to each execution processor by the manager. Convergence is determined when there is not a significant change of model parameters in consecutive iterations. At convergence, the output parameters (e.g., π'_{jk}) are normalized because the sum of the mixture's weights is no longer equal to 1 after parallel sampling.

3.2.2. Parallel Aspect-Sentiment Identification. The aspect-sentiment associations of the proposed product ontology are learnt from a big corpus of product comments periodically. As shown in Figure 5, Spark Workers first divide the big data into some Resilient Distributed Datasets (RDDs) that contain multiple data partitions for parallel processing. These RDDs are transformed by the corresponding functions (e.g., map, filter, collect, and reduce functions) to identify the aspect-sentiment associations. The transformation primitives such as the *Map* function conducts the necessary computations (e.g., parsing aspect-sentiment pairs from product comments), and the action primitives such as the *Reduce* function aggregates the intermediate computational results to generate the new RDDs (e.g., aggregated aspect-oriented sentiments for a product category) for further in-memory processing.

Figure 5 Parallel Aspect-Sentiment Identification on Spark [Color figure can be viewed at wileyonlinelibrary.com]



The purpose of aspect-sentiment identification is to identify all possible aspect-sentiment pairs, and determining their context-sensitive polarities based on a big volume of product comments. During the Map process, each product comment (document) within a RDD is scanned. Essentially, a virtual text window of size measured by ω_{win} words is constructed and is moved from left to right within a document. Within each text window, if an elementary aspect is identified, the adjacent seeding sentiments defined in sentiment lexicons such as Opinionfinder (Riloff et al. 2005) and SentiWordNet (Esuli and Sebastiani 2005) will be detected. If a seeding sentiment is not found, the proposed sentiment identification method will identify the adjacent adjective or adverb as a candidate sentiment indicator (Hu and Liu 2004, Subrahmanian and Recupero 2008). For the experiments reported in this study, we employed $\omega_{win} = 6$ since it was empirically tested to be an effective text windowing parameter (Lau et al. 2014). The proposed method takes into account the sentence boundary in that only the adjective (or adverb) co-located with an elementary aspect in the same sentence is extracted. After identifying each aspect-sentiment pair (a_i, s_i), the next step is to infer its context-sensitive polarity. In particular, we leverage the rich collective intelligence (e.g., the sheer volume of user-rated product reviews) to infer the context-sensitive polarity of each aspect-oriented sentiment. We make use of the distributional characteristics of big data in the following ways: (1) a positive product review is more likely to contain positive sentiments; (2) if an aspect-oriented sentiment repeatedly appears in many positive (negative) reviews, the sentiment is very likely to have a positive (negative) polarity.

Based on the well-known measure of Kullback–Leibler (KL) divergence, we develop the sentiment divergence (SD) measure to determine the polarity and the polarity strength of an aspect-oriented sentiment. As user-rated product reviews are readily available on the Internet (e.g., Amazon, JD, etc.), our proposed method does not require the intrusive human labeling process, and hence making it scalable for big data. The polarity $PO(t)$ and the associated polarity strength $PS(t)$ of an aspect-sentiment pair $t = (a_i, s_i)$ is defined as follows.

$$SD(t) = \Pr(pos|t) \times \log_2 \frac{\Pr(pos|t)}{\Pr(pos)} - \Pr(neg|t) \times \log_2 \frac{\Pr(neg|t)}{\Pr(neg)}, \quad (3)$$

$$NSD(t) = \frac{e^{SD(t)} - e^{-SD(t)}}{e^{SD(t)} + e^{-SD(t)}}, \quad (4)$$

$$PS(t) = \begin{cases} \dfrac{NSD(t) - \mu_{SD}}{1 - \mu_{SD}} & \text{if } NSD(t) > \mu_{SD} \\ -\left(\dfrac{|NSD(t)| - \mu_{SD}}{1 - \mu_{SD}}\right) & \text{if } NSD(t) < -\mu_{SD} \\ 0 & \text{otherwise} \end{cases}, \quad (5)$$

$$PO(t) = \begin{cases} pos & \text{if } PS(t) > 0 \\ neg & \text{if } PS(t) < 0 \\ neu & \text{otherwise} \end{cases}. \quad (6)$$

The conditional probability $\Pr(pos|t) = \frac{df(t_{pos})}{df(t)}$ is estimated by counting the number of positive reviews containing the aspect-sentiment pair $t = (a_i, s_i)$ given the number of reviews containing the same pair. In addition, $\Pr(pos) = \frac{|X^+|}{|X^+|+|X^-|}$ ($\Pr(neg) = \frac{|X^-|}{|X^+|+|X^-|}$) is the priori probability that a review is positive (negative); X^+ (X^-) is the set of positive (negative) reviews rated by users. The normalized sentiment divergence score $NSD(t) \in [-1, +1]$ is derived by using Equation 4. Moreover, Equation 5 is applied to filter noisy sentiments, where the filtering threshold μ_{SD} is empirically established. Finally, Equation 6 is applied to establish the polarity label of an aspect-oriented sentiment. The negation of an aspect-oriented sentiment is taken into account by the proposed method. For instance, if words such as "no," "not," "except," and so on captured in the negation lexicon is found within a virtual text window that contains the pair $t = (a_i, s_i)$, the sign of the corresponding polarity strength $PS(t)$ is negated. If more than one negation word is found in the same virtual text window, the flipping rule is applied to determine the final sign of the pair. The polarity strength of a high-level aspect (e.g., the exposure aspect of a camera) is determined by the average polarity strength of its constituent elementary aspects. Although some product comments such as those retrieved from Twitter or Sina Weibo do not contain user ratings, we can apply the pseudo-labeling method (Lau et al. 2012) to infer the polarity label of the comments first. Once the polarity label of a product comment is inferred, we can then apply the aforementioned method to extract the aspect-oriented sentiments.

3.3. Parallel Sentiment Analyzer

The function of the parallel sentiment analyzer is to conduct aspect sentiment classification tasks. Once a product ontology is learnt, aspect sentiment classification becomes straightforward because the product ontology can be utilized as an aspect-oriented sentiment lexicon. To estimate the polarity strength $PSP(p) \in [-1, +1]$ of a product p and the polarity strength $PSC(pc) \in [-1, +1]$ of a product category pc, we propose the following metrics. The sets X_p and AS_d represent all the reviews pertaining to a product p and the set of aspect-sentiment pairs found in a review d, respectively. Moreover, the set P_c represents the set of products pertaining to a product categorypc.

$$PSP(p) = \frac{\sum_{d \in X_p} \sum_{t \in AS_d} PS(t)}{|X_p| \times |AS_d|}, \quad (7)$$

$$PSC(pc) = \frac{\sum_{p \in P_c} PSP(p)}{|P_c|}. \quad (8)$$

To ensure efficient sentiment analysis against big data, we develop a parallel aspect sentiment classification algorithm as depicted in Figure 6. Each Map function parses a subset of product comments captured in a RDD in parallel. For each product comment d, the Map function first determines the product associated with the comment, and then identifies an aspect-oriented sentiment from each virtual text window by matching the pattern with that stored in our product ontology. If an aspect-oriented sentiment is found, its corresponding polarity and polarity strength are retrieved from the product ontology. Then, the Map function writes all the aspect-oriented sentiments pertaining to the current product and product category to memory. After the in-memory shuffling process performed by Spark (e.g., grouping parsed aspect-sentiment pairs by product), the Reduce functions will be initiated by the Spark Workers. Each Reduce function reads the intermediate processing results from main memory and performs the aggregation tasks. For example, the average sentiment score for each unique product is calculated. The average sentiment scores for some products are then accumulated for each product category. Finally, the sentiment score of a product, the sentiment score of a product category and the aspect-sentiment pairs of each product are written to a RDD, and subsequently stored in permanent storage (e.g., HDFS).

3.4. Product Demand Estimator

The function of the product demand estimator is to perform sentiment-based sales forecasting. We refer to the published work to identify a set of common features that are known to produce good forecasting performance (Ahn and Spangler 2014, Archak et al. 2011, Chevalier and Mayzlin 2006, Sonnier et al. 2011, Yu et al. 2012). These features include historical sales

Figure 6 The Map and the Reduce Functions for Parallel Aspect Sentiment Classification

```
Task Transformation

Function INITIALIZE()
    Ont ← ReadRDD ProductOnt;                        // read product ontology
end

Function MAP(docid id, doc d)
    AS ← new Array;                                  // initialize array
    n ← CountW(d);                                   // count no. of words in document
    p ← GetP(d,Ont);                                 // match product with Ontology
    for Idx = 1 to n
        TWin ← Get(Idx, ϖ_win, d);                   // build virtual text window
        if NOT sentence boundary
            if ((a_i,s_i) ∈ TWin) ∧ ((a_i,s_i) ∈ Ont); // aspect-sentiment found
                PO ← GetPO((a_i,s_i),Ont);           // get polarity from Ontology
                PS ← GetPS((a_i,s_i),Ont);           // get polarity strength from Ontology
                Append(AS,⟨(a_i,s_i),PO,PS⟩);        // save sentiment information
            end
        end
    end
    pcat ← GetPcat(p,Ont);                           // get product category
    emit(key pcat, tuple⟨p,id,AS⟩);                  // write to memory
end
```

```
Task Action

Function INITIALIZE()
    PC ← new associative Array;                      // array for product category
    CScore ← 0;                                      // initialize category score
    idx ← 0;                                         // initialize index
end

Function REDUCE(key pcat, tuple list ⟨p_1,id_1,AS_1⟩···⟨p_n,id_n,AS_n⟩)
    for each ⟨p_i,id_i,AS_i⟩ ∈ tuple list [⟨p_1,id_1,AS_1⟩···⟨p_n,id_n,AS_n⟩] do
            AScore ← PSP(p_i);                       // compute average sentiment score
            CScore ← CScore + AScore;                // accumulate category score
            if p_i ∈ PC[idx,1]                       // product is found in array
                Score ← PC[idx,2];                   // get current average sentiment score
                Score ← Score + AScore;              // accumulate new average score
                PC[idx,2] ← Score;                   // update array
            else
                append(PC,⟨p_i,AScore,AS_i⟩);        // insert new element to array
            end
            idx ← idx + 1;
    end
    CScore ← CScore ÷ idx;                           // computer average category score
    emit(key pcat, tuple ⟨CScore,PC⟩);               // write new RDDs
end
```

amounts, price of a product, search volume of a product, and the sentiment score of a product. As the search volume captured on Internet search engines may also reflect consumers' demands (Ahn and Spangler 2014, Archak et al. 2011), we access to the Google Trends[5] (or Baidu Index[6]) to retrieve the search volume related to a product. All the feature values are normalized before they are fed to a predictive model.

Moreover, we extend and integrate a parallel co-evolutionary genetic algorithm (PCGA) (Zhang et al. 2014) into the extreme learning machine (ELM) (Huang et al. 2004) to build the parallel co-evolutionary ELM (PELM) for the product demand estimator. ELM is a single hidden-layer feed forward neural network (SLFN). Suppose that there are N distinct samples (i.e., observations)$(\mathbf{x}_i, \mathbf{y}_i)$, where $\mathbf{x}_i = \langle x_{i1}, x_{i2}, \cdots x_{ij} \rangle \in R^j$ and $\mathbf{y}_i = \langle y_{i1}, y_{i2}, \cdots y_{ik} \rangle \in R^k$ are the input and the corresponding output vectors, respectively. A standard SLFN with L hidden nodes and activation functions $\mathrm{A}(\cdot)$ is formally defined by:

$$\sum_{i=1}^{L} \mathbf{o}_i \mathrm{A}(\mathbf{w}_i \bullet \mathbf{x}_h + b_i) = \mathbf{y}_h, h = 1, 2, \ldots N, \tag{9}$$

where $\mathbf{w}_i = \langle w_{i1}, w_{i2}, \cdots w_{ij} \rangle \in R^j$ is the input weight vector which connects the ith hidden neuron of the hidden layer and all the neurons of the input layer, and $\mathbf{o}_i = \langle o_{i1}, o_{i2}, \cdots o_{ik} \rangle \in R^k$ is the output weight vector which links the ith hidden neuron and all the neurons of the output layer. The term b_i represents the activation threshold of the ith neuron at the hidden layer, and $\mathbf{w}_i \bullet \mathbf{x}_h$ is the inner product of the vectors $\mathbf{w}_i$ and $\mathbf{x}_h$. If the SLFN can approximate the N samples with a zero error, it suggests that:

$$\sum_{i=1}^{L} \mathbf{o}_i \mathrm{A}(\mathbf{w}_i \bullet \mathbf{x}_h + b_i) = \mathbf{t}_h, h = 1, 2, \ldots N, \tag{10}$$

where $\mathbf{t}_h$ is the actual output (e.g., actual sales amount). The above N equations can be written in a compact format such as $\mathbf{AO} = \mathbf{T}$, where $\mathbf{A}$, $\mathbf{O}$, $\mathbf{T}$ represent the matrix of activation functions, the vector of output weights, and the vector of actual outputs, respectively. Its equivalent matrix format can be expressed as follows.

$$\begin{pmatrix} \mathrm{A}(\mathbf{w}_1 \bullet \mathbf{x}_1 + b_1) & \cdots & \mathrm{A}(\mathbf{w}_L \bullet \mathbf{x}_1 + b_L) \\ \vdots & \ddots & \vdots \\ \mathrm{A}(\mathbf{w}_1 \bullet \mathbf{x}_N + b_1) & \cdots & \mathrm{A}(\mathbf{w}_L \bullet \mathbf{x}_N + b_L) \end{pmatrix}_{N \times L} \times \begin{pmatrix} \mathbf{o}_1 \\ \vdots \\ \mathbf{o}_L \end{pmatrix}_L^T = \begin{pmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_N \end{pmatrix}_N^T. \tag{11}$$

It has been proved that the parameters of the hidden layer (e.g., number of neurons, input weights of neurons, and activation thresholds) can be assigned randomly if the activation function is infinitely differentiable in any interval (Huang et al. 2006). Thereby, finding the output weights is as simple as deriving the least-square solution for the linear system as defined in Equation 11, and a unique solution can be determined by using the Moore–Penrose generalized inverse (Huang et al. 2006). ELMs have demonstrated to achieve good generalization performance and improve learning efficiency when compared to artificial neural networks with back-propagation (BP) learning (Huang et al. 2004, 2006). Since the input weights and other parameters of the hidden layer of an ELM are randomly assigned, the outputs generated by a trained ELM could be different given the same input sample. This may be a problem for its application to sales prediction, and so an extended ELM (ELME) has been proposed (Sun et al. 2008). It is shown that ELME can improve the performance of sales prediction by computing the mean of the predicted values derived from a large number of independent ELMs (e.g., between 100 and 1000 ELMs) (Sun et al. 2008). Though ELME may enhance sales prediction performance, such an improvement usually requires the invocation of a large number of ELMs to derive an average prediction value. Nevertheless, invoking a large number of ELMs is computationally expensive under a big data environment (e.g., predicting hundred thousands of items).

Since genetic algorithms (GAs) can generally produce near optimal search results (Goldberg 1989, Lau et al. 2006), we extend a parallel co-evolutionary genetic algorithm (PCGA) (Zhang et al. 2014) to search for near optimal input weights and other parameters of the hidden layer during training time. In fact, the co-evolutionary approach simulates the real-world biological processes in that populations of species distributed in different locations (e.g., islands) can co-evolve together (Olsson 2001, Zhang et al. 2014). Occasionally, some species of a location may migrate to other locations, and hence driving the co-evolution of other populations. The advantage of the co-evolutionary approach is that a large search space is divided into subspaces for a parallel and diversified search (e.g., via a cluster of computing nodes), which improves both the efficiency and the effectiveness of the entire evolution process (Olsson 2001). Such an efficiency boost is particularly important while applying heuristic search methods to big data (Zhang et al. 2014). At testing time, the near optimal ELM is applied to sales prediction without the need of running it many times because all the parameters of the hidden layer have been tuned and fixed during training time.

4. Experiments and Results

4.1. Experimental Procedures and Datasets

We applied our crawlers and external APIs to retrieve product comments from various social media sources. The details of our big dataset are depicted in Table 2. We simply adopted 30 Amazon high-level product categories to build our product taxonomy and applied them to categorize all product comments written in English. On the other hand, we adopted the 41 high-level product categories of JD.com to categorize all product comments written in Chinese. For product comments that were not explicitly attached to a product (e.g., tweets), we applied the NER tool from our data pre-processing layer to identify the specific product that a comment referred to.

Table 2 The Bilingual Datasets for System Evaluation

Source	No. of reviews	No. products	No. product categories	Period
Amazon	15,675,261	852,854	30	2008–2013
Epinions	309,193	137,446	30	2008–2012
Newegg	434,218	155,302	30	2011–2014
Twitter	18,245,628	362,115	30	2009–2012
Dianping (Chinese)	2,245,116	103,356	41	2009–2014
Taobao (Chinese)	6,137,255	422,169	41	2009–2014
JD (Chinese)	1,328,303	346,082	41	2009–2014
Sina Weibo (Chinese)	16,193,712	215,293	41	2009–2014

The experimental setting was a cluster of 21 commodity computers, where one node was assigned as the maser node, and the remaining nodes were the worker nodes. Each computing node was equipped with a dual 6-core Intel Xeon E5-2620, 2.00 GHz processor with 64 GB RAM and 1TB hard disk, running 64-bit Ubuntu Linux. In terms of software, we used Apache Spark version 1.6.0 with Java 1.6 installed. We also tested the serial time of running our prototype system under a single host environment. The single host was a Dell PowerEdge M910 server with 10-core Xeon 2.4 GHz processors, 64 GB RAM, and 16 TB hard disk.

4.2. Experiments for Aspect Sentiment Extraction

Since the product taxonomy $R_{P \times P}$ is directly retrieved from existing e-Commerce sites, our main focus is to evaluate the proposed computational method for learning the aspect-sentiment relation $R_{A \times S \times PO}$ of the product ontology. We divided our corpus to the English subset and the Chinese subset, and performed two series of experiments separately. For each experimental run, we used all available reviews pertaining to a product category to learn the aspects and sentiments for that product category. We adopted the measure of precision-at-k (McFee and Lanckriet 2010) to evaluate the effectiveness of aspect-sentiment extraction because it was not practical to develop the complete ground truth of aspect-sentiments for a product domain. We invited 20 domain experts from a large e-Commerce firm to evaluate the correctness of the aspect-sentiments extracted by the PASAD system. These experts were allocated to 10 assessment groups with each group containing 2 members. For a practical evaluation exercise, we selected the top 10 high-level aspects (concepts) measured in terms of concept entropy (Lau et al. 2015, Wang et al. 2014) from each product category to be inspected by domain experts. For each high-level aspect, the top 5, 10, and 15 elementary aspects ranked by the generation probabilities were inspected by experts to assert if the corresponding high-level aspect was extracted correctly. Each pair of domain experts was randomly assigned to evaluate the aspect-sentiments (i.e., high-level aspects, elementary aspects, and the associated sentiment polarities) of a product category. Only if both experts agreed on an assessment, would the particular aspect-sentiment be regarded as correctly learnt by a system.

By searching existing literature, we only found very few studies about automated learning of product ontologies with embedded sentiments. Our first baseline system was the OBPRM system that used LDA for aspect mining (Lau et al. 2014). The second and the third baselines employed the FL-LDA and the UFL-LDA models, respectively (Wang et al. 2014). For a comparative evaluation, we extended all these methods by using the parallel LDA method (Wang et al. 2009). The number of pre-defined topics of each LDA-based model was established based on the perplexity measure (Lau et al. 2015). The proposed PASAD system was implemented under Spark, and all the baseline systems were implemented under Hadoop though all systems operated under the same cluster. Table 3 shows the top 3 aspects of the digital camera product category learnt by the proposed PASAD system, and Table 4 shows the comparative average precision-at-k achieved by all systems under testing. According to Table 4, the proposed PASAD system achieves the best precision-at-k because the proposed parallel GP-HDP topic modeling method is more effective for learning representative aspects based on the underlying latent aspect distributions of a dataset. There is no need to manually and artificially choose the number of topics for the GP-HDP model.

Moreover, the proposed Spark-based implementation is more efficient than the Hadoop-based baselines even though the ontology learning tasks were mainly running in batch mode. The PASAD system completed the aspect sentiment extraction tasks by using 15,609 seconds (English) and 19,033 seconds (Chinese), respectively. If the circular queue-based workload balancing scheme is not used, the efficiency of the PASAD system will drop by 32%. The proposed system is faster than the best baseline system (FL-LDA) by 40.5% (English) and 34.6% (Chinese), respectively. Efficiency improvement over the baselines is brought by the in-memory data analytics (e.g., parallel Gibbs sampling) on Spark when compared to the disk-based data shuffling operations conducted by Hadoop. The scalability of various systems for the aspect sentiment extraction tasks toward the English review corpus is plotted in Figure 7a. Basically, we observe a linear scalability of all systems with respect to the growing size of the review corpus. On the other hand, the serial version of the PASAD system took 125 hours and 149 hours for completing the aspect sentiment extraction tasks for the English and the Chinese corpora, respectively, on our single host server. Obviously, the proposed big data analytics methodology considerably outperforms the traditional single host approach.

4.3. Experiments for Aspect Sentiment Classification

Given numerous user-rated online product comments (e.g., Amazon or JD reviews), we can make use of these readily available ground-truths to evaluate the effectiveness of parallel aspect sentiment classification of PASAD. The basic assumption is that the set of aspect-oriented sentiments embedded in a review should lead to the final polarity label of the corresponding review. The polarity score of a review is the average aspect-oriented sentiment score of that review. If the polarity score of a review is greater than zero, the review is classified as positive; otherwise, it is taken as negative. We separately evaluated a

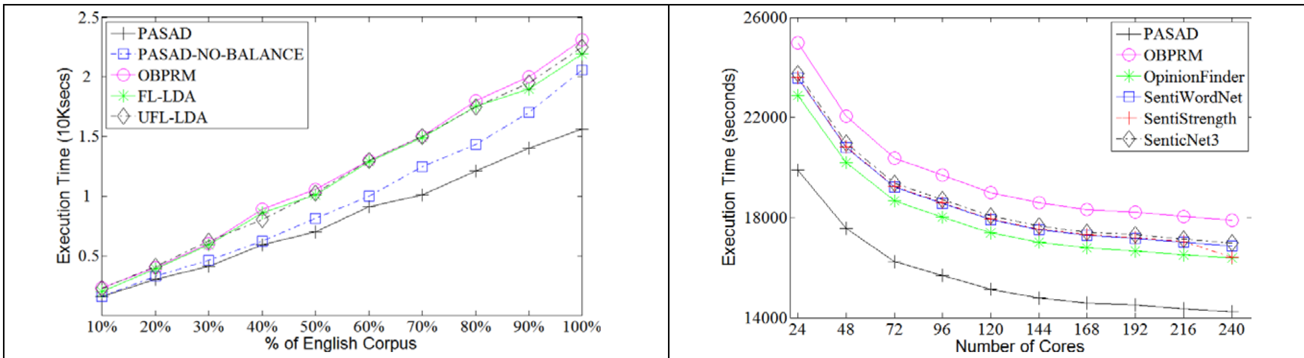
Table 3 Top 3 Aspect-Sentiments of the Digital Camera Product Category Learnt by PASAD

Aspect-1: display			Aspect-2: recording			Aspect-3: exposure		
E. Aspect	Sent.	PO	E. Aspect	Sent.	PO	E. Aspect	Sent.	PO
Display	Large	Pos	Recording	Handy	Pos	Iso	Available	Pos
View	Full	Pos	Sensor	Flexible	Pos	Balance	Well	Pos
Screen	Small	Neg	Image	Vague	Neg	Exposure	Good	Pos
Monitor	Unclear	Neg	Support	Poor	Neg	Shutter	Fast	Pos
Lcd	Bright	Pos	Card	Expensive	Neg	Speed	Slow	Neg
Picture	Clear	Pos	Speed	Fast	Pos	Button	Sensitive	Pos
Light	Dim	Neg	Memory	Internal	Pos	Range	Small	Neg
Screen	Wide	Pos	Image	Clear	Pos	Mode	Quick	Pos

Table 4 Comparative Performance of Aspect Sentiment Extraction

System	Precision@5	Precision@10	Precision@15	Time (English)	Time (Chinese)
PASAD (with workload balancing)	0.891	0.813	0.754	15,609	19,033
PASAD (without workload balancing)	0.891	0.813	0.754	20,574	24,933
OBPRM	0.742	0.671	0.583	23,125	26,843
FL-LDA	0.815	0.755	0.691	21,932	25,608
UFL-LDA	0.792	0.711	0.665	22,495	26,045

Figure 7 The Scalability of Aspect-Oriented Sentiment Analysis [Color figure can be viewed at wileyonlinelibrary.com]



(a) Parallel Aspect Extraction

(b) Sentiment Classification with Varying Cores

system's performance for positive (star ratings 4 and 5) and negative (star ratings 1 and 2) review classifications, and then computed the average performance of the system across product categories. For this series of experiments, we only employed the Amazon and the JD reviews that contained user-annotated labels (i.e., the star ratings). The performance metrics that we used included precision $=\frac{a}{a+b}$, recall $=\frac{a}{a+c}$, and $F_{\beta=1}=\frac{(1+\beta^2)\times precision\times recall}{\beta^2\times precision+recall}=\frac{2a}{2a+b+c}$, whereas a, b, c, d represent the number of correctly classified positive (negative) reviews, the number of misclassified non-positive (non-negative) reviews, the number of misclassified positive (negative) reviews, and the number of correctly classified non-positive (non-negative) reviews, respectively.

The baseline systems included a parallelized version of the OBPRM system (Lau et al. 2014) and well-known lexicon-based sentiment analysis methods such as OpinionFinder (Riloff et al. 2005), SentiWordNet (Esuli and Sebastiani 2005), SentiStrength (Thelwall et al. 2012), and SenticNet-3 (Cambria et al. 2014). For Chinese sentiment analysis, we employed the HowNet (Dong and Dong 2003) Chinese word lexicon and the Chinese translation of the OpinionFinder lexicon. All baseline systems operated under Hadoop, whereas the PASAD system operated under Spark. The sentiment classification performance and the execution time of all systems are shown in Table 5. Again, the proposed PASAD system outperforms other baseline systems. The PASAD system predicts the polarity labels of reviews more accurately because it can perform context-sensitive sentiment classification by using aspect-oriented sentiments learnt during the product ontology building stage. Although the concept-based SenticNet-3 lexicon (Cambria et al. 2014) captures semantically rich descriptions about aspects and sentiment indicators, the polarities of sentiments are not context-sensitive. For instance, the sentiment "small" has a negative polarity strength of "−0.113" in SenticNet-3. Nevertheless, the sentiment "small" has a positive polarity for consumer electronics such as "my small size notebook is ideal for traveling." As a result, context-free lexicon-based methods tend to produce quite a number of errors for review polarity classification in some product categories.

Table 5 Comparative Performance of Parallel Aspect Sentiment Classification

System	Precision	Recall	$F_{\beta=1}$	Time (sec.)
Amazon (English)				
PASAD	0.852	0.877	0.864	14,246
OBPRM	0.799	0.804	0.801	17,094
Opinionfinder	0.669	0.717	0.692	16,382
SentiWordNet	0.685	0.739	0.711	16,503
SentiStrength	0.687	0.746	0.715	16,411
SenticNet3	0.717	0.789	0.752	16,817
JD (Chinese)				
PASAD	0.806	0.812	0.809	1992
OBPRM	0.727	0.732	0.729	2390
Opinionfinder	0.597	0.638	0.618	2291
HowNet	0.608	0.652	0.629	2303

On the other hand, the PASAD system is the most efficient one because of the Spark-based in-memory sentiment matching and polarity prediction operations when compared to the disk-based sentiment score shuffling on Hadoop. It takes around 1 millisecond and 1.5 milliseconds for the PASAD system to analyze the sentiments of each Amazon review and JD review, respectively. Figure 7b shows the scalability of all systems with respect to varying number of cores for the sentiment classification tasks of the Amazon corpus. Initially, when more cores are used in a cluster, the computational time of a system is considerably reduced. However, when it reaches a certain threshold (e.g., 168 cores), the reduction in computational time becomes smaller. Such a reduction is perhaps caused by the increasing communication overheads among the computing nodes of a large cluster. In sum, our experiments demonstrate that the proposed PASAD system is able to effectively and efficiently perform large-scale sentiment analysis.

4.4. Experiments for Sales Forecasting

To evaluate the impact of aspect-oriented sentiments on sales forecasting performance, we used both the publicly available Amazon dataset and a proprietary dataset extracted from a large Chinese e-Commerce firm in these experiments. In particular, we used the weekly sales data (the sales ranks in case of Amazon products) and the corresponding weekly review sentiments extracted from March 2011 to February 2013 (i.e., a total of 109 weekly sales amounts) for system evaluation. Aside from the PELM experimental predictive model, we also implemented several baseline predictive models, namely linear regression (LR), support vector regression (SVR) (Vapnik 2000), and extended extreme learning machine (ELME) (Sun et al. 2008). Similar to previous studies (Ahn and Spangler 2014, Archak et al. 2011), the proposed PELM model and the aforementioned baseline models utilized features such as historical sales (e.g., sales of the previous week, moving average of the past 3 weeks, and moving average of the past 4 weeks), search volume, price, and sentiment scores extracted from the previous periods (e.g., sentiment score of the previous week, moving average of the past 3 weeks, and moving average of the past 4 weeks) to predict the sales performance of each product in the following period (e.g., week). However, we exploited aspect-oriented sentiments and the PELM model to enhance sales forecasting performance in this study. For further comparative evaluation, we also implemented the ARSA model (Liu et al. 2007, Yu et al. 2012) which only used historical data and document-level sentiments as predictive features.

We employed two common performance measures, namely root mean squared error (RMSE) and mean absolute percentage error (MAPE) in this study. In general, the RMSE measure is more sensitive to large prediction errors since the difference between each pair of predicted and actual values is squared before the summation. These measures are formally defined as follows:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} (y_i - t_i)^2}{N}}, \tag{12}$$

$$\mathrm{MAPE} = \frac{\sum_{i=1}^{N} \left| \frac{(y_i - t_i)}{t_i} \right|}{N} \times 100\%, \tag{13}$$

where y_i and t_i represent the predicted and the actual values, respectively. The term N denotes the total number of samples under evaluation. Within each product category, there are fast-moving or slow-moving products which can be trivially predicted (e.g., always a low sales volume). Therefore, we chose products that were relatively fast-moving and more likely having changing sales volume across weeks. Moreover, products with very few consumer reviews during the evaluation period, or without the corresponding search volume data archived at Google or Baidu were also filtered out. At the end, we successfully established a big evaluation dataset which contained 11,428 Amazon products and the corresponding 1,251,828 reviews across 26 product categories, and 8115 products and the corresponding 451,113 reviews across 38 product categories for the Chinese e-Commerce firm. Similar to the training and the testing split of the previous studies (Lu et al. 2012, Wong and Guo 2010), we used the first half of the time series (i.e., 55 weeks) of a product for model training, and the second half of the time series (i.e., 54 weeks) as hold-out samples for testing.

For the SVR model, we adopted the radial basis function (RBF) as the kernel function with controlling parameter $\sigma = 0.2$ (Lu et al. 2012). For the ELME model, we invoked each ELM 100 times to obtain an average prediction value for each testing sample (Sun et al. 2008). Moreover, the parameters $K = 4$, $p = 7$, and $q = 1$ were set for the ARSA model (Yu et al. 2012). As for the PELM model, we applied a real-value gene encoding scheme (Zhang et al. 2014) to each chromosome to capture various parameters of the hidden layer. Moreover, the fitness function was defined by using the RMSE measure, and the Roulette wheel selection method (Goldberg 1989) was applied to choose relatively fitter chromosomes from the current generation to produce chromosomes of the next generation. The size of each population was set to 100, and the initial mutation rate and the crossover rate were set to 0.82 and 0.05, respectively (Zhang et al. 2014). Finally, the migration rate of the co-evolution process was set to 10, which suggested that the fittest chromosome of a population was migrated to other populations after every 10 generations to drive the evolution processes of other populations. For each predictive model, we performed two sets of runs, a run with the sentiment feature and another run without using the sentiment feature. The sales forecasting tasks were conducted across product categories in parallel. The average forecasting performance and the prediction time of various models under Spark are tabulated in Table 6.

According to our experimental results tabulated in Table 6, sales forecasting performance is improved across all predictive models and datasets if the sentiment feature is used. For the PELM model, it achieves

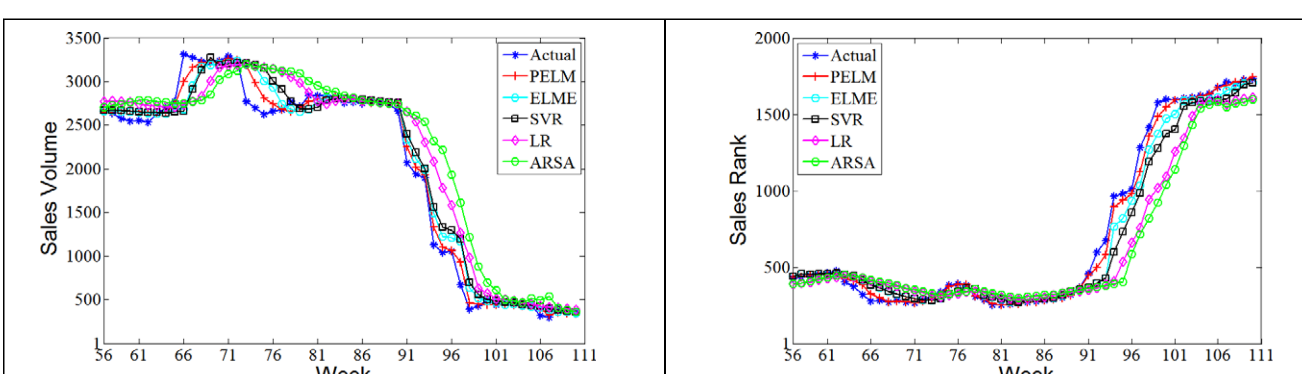
Table 6 Comparative Performance of Weekly Sales Forecasting

	Average RMSE			Average MAPE			Prediction time (S)	
Model	With sentiments	Without sentiments	Improve	With sentiments	Without sentiments	Improve	Training	Testing
Amazon dataset								
LR	514.7	555.3	+7.9%	48.1%	51.1%	+6.2%	142.9	59.4
ARSA	548.1	598.5	+9.2%	50.6%	54.8%	+8.3%	137.1	57.1
SVR	271.5	302.2	+11.3%	23.0%	25.2%	+9.6%	234.3	77.1
ELME	268.5	295.9	+10.2%	22.8%	24.9%	+9.2%	228.6	742.8
PELM	239.6	269.1	+12.3%	20.7%	22.9%	+10.6%	282.8	74.3
Proprietary Chinese Dataset								
LR	609.7	651.4	+6.8%	55.7%	59.0%	+5.9%	103.3	53.8
ARSA	602.2	648.2	+7.6%	55.2%	58.6%	+6.2%	101.8	51.9
SVR	314.4	346.7	+10.3%	26.1%	28.2%	+8.0%	192.1	73.8
ELME	311.1	343.8	+10.5%	25.9%	28.1%	+8.5%	188.3	537.9
PELM	281.2	315.2	+12.1%	23.9%	26.1%	+9.2%	234.4	69.2

an average improvement of 12.2% in terms of RMSE and 9.9% in terms of MAPE when aspect-oriented sentiments are applied to sales forecasting. As for prediction accuracy, the proposed PELM model outperforms all baselines in terms of both RMSE and MAPE in both datasets. The PELM model performs better than the best baseline (i.e., the ELME model) by an average of 11.3% in terms of RMSE. Unlike the classical time series-based predictive models that mainly consider historical sales data, the proposed model not only takes into account the trend of a time series but also the subjective views of consumers toward products as captured in the aspect-oriented sentiment scores. Even though consumers' preferences may change over time (e.g., due to newer and more advanced products are introduced to the market), the weekly sentiments are able to capture such preferential changes. As a result, the predictive models that incorporate the sentiment feature can produce more accurate sales forecasting when compared to the models that exclude such a feature. The PELM model achieves the best sales forecasting performance, and the possible reason is that the PELM model is more effective to approximate the complex, nonlinear relationship between the set of features (e.g., historical sales, sentimental factor, trend, etc.) and the actual sales when compared to other predictive models. All predictive models seem performing better for the Amazon dataset. There are two possible reasons. First, the weekly fluctuation of actual sales volume is larger than that of sales rank, and so making it more challenging to predict actual sales volume. Second, it is more difficult to accurately extract aspect-oriented sentiments from the Chinese reviews of the proprietary dataset due to errors of the Chinese word segmentation process.

Although the PELM model requires slightly more training time than other models, its testing time is comparable to other baseline models. On average, it only takes 1.2 minutes to complete 617,112 predictions for the Amazon test data (i.e., 0.1 millisecond per prediction). The typical predictions produced by various models for the sample product Dell U2410 LCD monitor are plotted in Figures 8a (Sales of the Chinese e-Commerce firm) and b (Amazon Sales Rank), respectively. It should be noted that a smaller Amazon sales rank implies a larger sales volume. In addition, we also conducted sales prediction

Figure 8 Typical Predictions of Various Models for the Dell U2410 LCD monitor [Color figure can be viewed at wileyonlinelibrary.com]



(a) Sales on the Chinese e-Commerce Firm

(b) Amazon Sales Rank

Table 7 Comparative Performance of Monthly Sales Forecasting

	Average RMSE			Average MAPE			Prediction time (S)	
Model	With sentiments	Without sentiments	Improve	With sentiments	Without sentiments	Improve	Training	Testing
Amazon Dataset								
LR	301.2	322.9	+7.2%	25.1%	26.5%	+5.6%	70.5	11.5
ARSA	325.5	353.2	+8.5%	27.1%	28.8%	+6.3%	67.1	11.2
SVR	163.3	182.8	+11.9%	12.9%	14.1%	+9.3%	119.5	12.4
ELME	159.4	178.7	+12.1%	12.5%	13.8%	+10.4%	118.7	120.4
PELM	141.8	160.1	+12.9%	11.3%	12.5%	+10.6%	129.0	12.2
Proprietary Chinese Dataset								
LR	349.5	372.5	+6.6%	28.2%	29.7%	+5.3%	53.3	8.8
ARSA	352.2	378.4	+7.4%	28.6%	30.4%	+6.3%	50.8	8.6
SVR	174.4	195.2	+11.9%	13.4%	14.7%	+9.7%	77.4	9.1
ELME	171.3	191.5	+11.8%	13.0%	14.2%	+9.2%	76.1	88.8
PELM	153.3	172.3	+12.4%	12.1%	13.3%	+9.9%	88.8	8.9

experiments based on monthly sales data (the first 20 sales months for training and the last 4 sales months for testing). The experimental results are tabulated in Table 7. The sentiment feature still consistently improves sales forecasting performance across predictive models and datasets, and the PELM model outperforms other baselines. In general, a better prediction accuracy is observed for the monthly prediction tasks. The possible explanation is that the sudden changes of consumer demands across weeks are absorbed into a monthly average. Accordingly, it is relatively easier for a predictive model to predict the trend of a smoothed monthly time series.

5. Conclusions

Though many studies about supply chain management and sales forecasting have been done, research on designing big data analytics methodologies for improving sales forecasting is seldom reported in existing literature. The main contributions of our research are twofold. First, we design a novel big data analytics methodology named PASAD that is underpinned by a parallel aspect-oriented sentiment analysis method for mining consumer sentiments from the big data of online product comments. Second, we design and empirically evaluate a sentiment enhanced sales forecasting method that is empowered by the parallel co-evolutionary extreme learning machine. Based on real-world big datasets, our experimental results show that the proposed parallel aspect-oriented sentiment analysis method can effectively and efficiently mine consumer sentiments from big data. Moreover, sales forecasting performance is improved across predictive models and datasets by taking into account the consumer sentiments mined from big data. The PELM predictive model empowered by sentiment feature outperforms the best baseline model by 12.2% in terms of RMSE. The managerial implication of our work is obvious because firms can apply the proposed big data analytics methodology to enhance their sales forecasting performance. As a result, the problem of under/overstocking is alleviated, and customer satisfaction is raised. Ultimately, it enables firms to achieve sustainable competitive advantages. In the future, we will extend the parallel topic model by incorporating lifelong learning strategy to improve the quality of aspect extraction from big data. Moreover, parallel ensemble model for sales forecasting will be examined to bootstrap both the efficiency and effectiveness of real-time sales forecasting.

Acknowledgments

The authors thank Mr. Zhifeng Xing, the senior manager at the Big Data Division of JD.com for helpful advice and providing some evaluation data. The authors also appreciate the editors' and the anonymous reviewers' valuable comments. The first two authors are co-first authors, and Wei Xu is the corresponding author. This work was partially supported by grants from the Research Grants Council of Hong Kong SAR, China (Project: CityU 11502115 and Project: CityU 11525716), the NSFC Basic Research Program (Project: 71671155), and the City University of Hong Kong Shenzhen Research Institute, Shenzhen, P. R. China.

[Correction added on 3 January 2018, after first online publication: additional funding information has been inserted in the Acknowledgments section.]

Notes

[1]http://spark.apache.org/
[2]http://quantum.is.cityu.edu.hk/pasad/home.htm
[3]http://hadoop.apache.org/
[4]http://ictclas.org/
[5]trends.google.com/
[6]http://index.baidu.com/

References

Ahn, H., W. Spangler. 2014. Sales prediction with social media analysis. Proceedings of the 2014 IEEE SRII Global Conference, pp. 213–222.

Archak, N., A. Ghose, P. Ipeirotis. 2011. Deriving the pricing power of product features by mining consumer reviews. *Management Sci.* **57**(8): 1485–1509.

Bollen, J., H. Maoa, X. Zengb. 2011. Twitter mood predicts the stock market. *J. Comput. Sci.* **2**(1): 1–8.

Boyaci, T., G. Gallego. 2004. Supply chain coordination in a market with customer service competition. *Prod. Oper. Manag.* **13**(1): 3–22.

Breiter, A., A. Huchzermeier. 2015. Promotion planning and supply chain contracting in a high-low pricing environment. *Prod. Oper. Manag.* **24**(2): 219–236.

Cambria, E., D. Olsher, D. Rajagopal. 2014. SenticNet 3: A common and common-sense knowledge base for cognition-driven sentiment analysis. Proceedings of the 2014 AAAI Conference, pp. 1515−1521.

Canny, J., H. Zhao. 2013. Big data analytics with small footprint: Squaring the cloud. Proceedings of the 19th ACM SIGKDD Conference, pp. 95−103.

Chen, Z., B. Liu. 2014. Mining topics in documents: Standing on the shoulders of big data. Proceedings of the 20th ACM SIGKDD Conference, pp. 1116−1125.

Chen, Z., A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, R. Ghosh. 2013. Exploiting domain knowledge in aspect extraction. Proceedings of the 2013 EMNLP Conference, pp. 1655−1667.

Chen, Z., A. Mukherjee, B. Liu. 2014. Aspect extraction with automated prior knowledge learning. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pp. 347−358.

Cheng, D., Y. Liu. 2014. "Parallel Gibbs sampling for hierarchical Dirichlet processes via Gamma processes equivalence. Proceedings of the 20th ACM SIGKDD Conference, pp. 562−571.

Chevalier, J. A., D. Mayzlin. 2006. The effect of word of mouth on sales: Online book reviews. *J. Mark. Res.* **43**(3): 345–354.

Das, S., M. Chen. 2007. Yahoo! for Amazon: Sentiment extraction from small talk on the web. *Management Sci.* **53**(9): 1375–1388.

David, A., E. Adida. 2015. Competition and coordination in a two-channel supply chain. *Prod. Oper. Manag.* **24**(8): 1197–1374.

de Treville, S., N. Schürhoff, L. Trigeorgis, B. Avanzi. 2014. Optimal sourcing and lead-time reduction under evolutionary demand risk. *Prod. Oper. Manag.* **23**(12): 2103–2117.

Ding, X., B. Liu. 2007. The utility of linguistic rules in opinion mining. Proceedings of the 30th Annual International ACM SIGIR Conference, pp. 811–812.

Ding, X., B. Liu, P. Yu. 2008. A holistic lexicon-based approach to opinion mining. Proceedings of the 2008 International Conference on Web Search and Data Mining, February 11−12, 2008, pp. 231−239.

Dong, Z., Q. Dong. 2003. HowNet - a hybrid language and knowledge resource. Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering, pp. 820–824.

Doulkeridis, C., K. Nørvåg. 2014. A survey of large-scale analytical query processing in MapReduce. *VLDB J.* **23**: 355–380.

Gandomi, A., M. Haider. 2015. Beyond the hype: Big data concepts, methods, and analytics. *Int. J. Inf. Manag.* **35**(2): 137–144.

Geman, S., D. Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian relation of images. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**: 721–741.

Gep, B., P. Da. 1970. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *J. Am. Statist. Assoc.* **65**(332): 1509–1526.

Goes, B. P. 2014. Editor's comments: Big data and IS research. *MIS Q.* **38**: iii–viii.

Goldberg, D. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Boston, Massachusetts.

Hu, M., B. Liu. 2004. Mining and summarizing customer reviews. Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177.

Huang, G. B., Q. Y. Zhu, C. K. Siew. 2004. Extreme learning machine: a new learning scheme of feedforward neural networks. Proceedings of the Joint Conference on Neural Networks, pp. 985–990.

Jindal, N., B. Liu. 2008. Opinion spam and analysis. Proceedings of the 2008 International Conference on Web Search and Web Data Mining, pp. 219−229.

Lau, R. Y. K., M. Tang, O. Wong, S. Milliner, Y. Chen. 2006. An evolutionary learning approach for adaptive negotiation agents. *Int. J. Intell. Syst.* **21**(1): 41–72.

Liu, B. 2012. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, Williston.

Liu, Y., X. Huang, A. An, X. Yu. 2007. ARSA: a sentiment-aware model for predicting sales performance using blogs. Proceedings of the 30th ACM SIGIR Conference, pp. 607−614.

Liu, X., J. Zeng, X. Yang, J. Yan, Q. Yang. 2015. Scalable parallel EM algorithms for latent Dirichlet allocation in multi-core systems. Proceedings of the 24th WWW Conference, pp. 669−679.

Maynard, D., V. Tablan, C. Ursu, H. Cunningham, Y. Wilks. 2001. Named entity recognition from diverse text types. Proceedings of the 2001 Conference on Recent Advances in NLP, Bulgaria.

McAfee, A., E. Brynjolfsson. 2012. Big Data: The Management Revolution. *Harvard Business Review*. Available at https://hbr.org/2012/10/big-data-the-management-revolution/ar

McFee, B., G. Lanckriet. 2010. Metric learning to rank. Proceedings of the 27th International Conference on Machine Learning, pp. 775−782.

Mei, Q., X. Ling, M. Wondra, H. Su, C. Zhai. 2007. Topic sentiment mixture: modeling facets and opinions in weblogs. Proceedings of the 16th World Wide Web Conference, pp. 171−180.

Newman, D., P. Smyth, M. Welling, A. U. Asuncion. 2007. Distributed inference for latent dirichlet allocation. Advances in Neural Information Processing Systems, pp. 1081−1088.

Ramirez-Gallego, S., S. Garcia, H. Mourino-Talin, D. Martinez-Rego, V. Bolon-Canedo, A. Alonso-Betanzos, J. M. Benitez, F. Herrera 2015. Distributed entropy minimization Discretizer for big data analysis under Apache Spark. IEEE Trustcom/BigDataSE Conference, pp. 33−40.

Riloff, E., T. Wilson, P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, C. Cardie, S. Patwardhan. 2005. Opinionfinder: A system for subjectivity analysis. Proceedings of the 2005 HLTC/EMNLP Conference, pp. 34–35.

Si, J., A. Mukherjee, B. Liu, S. Pan, Q. Li, H. Li. 2014. Exploiting social relations and sentiment for stock prediction. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), pp. 1139−1145.

Titov, I., R. McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. Proceedings of the 46th Annual Meeting of the ACL, pp. 308−316.

Esuli, A., F. Sebastiani. 2005. Determining the semantic orientation of terms through gloss classification. Proceedings of the 14th ACM CIKM Conference. Bremen, Germany, pp. 617–624.

Gruber, T. 1993. A translation approach to portable ontology specifications. *Knowl. Acquisition* **5**(2): 199–220.

Huang, G. B., Q. Y. Zhu, C. K. Siew. 2006. Extreme learning machine: Theory and applications. *Neurocomputing* **70**(1–3): 489–501.

Kalfoglou, Y., M. Schorlemmer. 2003. Ontology mapping: The state of the art. *Knowl. Eng. Rev.* **18**(1): 1–31.

Lau, R. Y. K., J. L. Zhao, G. Chen, X. Guo. 2016. Big data commerce. *Inf. Manag.* **53**: 929–933.

Lau, R. Y. K., J. L. Zhao, W. Zhang, Y. Cai, E. Ngai. 2015. Learning context-sensitive domain ontologies from Folksonomies: A cognitively motivated method. *INFORMS J. Comput.* **27**(3): 561–578.

Lau, R. Y. K., C. Li, S. Liao. 2014. Social analytics: Learning fuzzy product ontologies for aspect-oriented sentiment analysis. *Decis. Support Syst.* **65**: 80–94.

Lau, R. Y. K., S. Liao, K. F. Wong, K. W. Chiu. 2012. Web 2.0 environmental scanning and adaptive decision support for business mergers and acquisitions. *MIS Q.* **36**(4): 1239–1268.

Liu, B. 2015. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press, New York, NY.

Lu, C., T. Lee, C. Lian. 2012. Sales forecasting for computer wholesalers: A comparison of multivariate adaptive regression splines and artificial neural networks. *Decis. Support Syst.* **54**: 584–596.

Olsson, B. 2001. Co-evolutionary search in asymmetric spaces. *Inf. Sci.* **133**(3): 103–125.

Rui, H., Y. Liu, A. Whinston. 2013. Whose and what chatter matters? The effect of tweets on movie sales. *Decis. Support Syst.* **55**(4): 863–870.

Sonnier, G., L. McAlister, O. Rutz. 2011. A dynamic model of the effect of online communications on firm sales. *Market. Sci.* **30**(4): 702–716.

Subrahmanian, V., D. Recupero. 2008. AVA: Adjective-verb-adverb combinations for sentiment analysis. *IEEE Intell. Syst.* **23**(4): 43–50.

Sun, Z. L., T. M. Choi, K. F. Au, Y. Yu. 2008. Sales forecasting using extreme learning machine with applications in fashion retailing. *Decis. Support Syst.* **46**: 411–419.

Tang, S., H. Gurnani, D. Gupta. 2014. Managing disruptions in decentralized supply chains with endogenous supply process reliability. *Prod. Oper. Manag.* **23**(7): 1198–1211.

Teh, Y. W., M. I. Jordan, M. J. Beal, D. M. Blei. 2006. Hierarchical Dirichlet processes. *J. Am. Statistic. Assoc.* **101**(476): 1566–1581.

Thelwall, M., K. Buckley, G. Paltoglou. 2012. Sentiment strength detection for the social Web. *J. Am. Soc. Inform. Sci. Technol.* **63**(1): 163–173.

Vapnik, V. N. 2000. *The Nature of Statistical Learning Theory*. Springer, New York.

Wang, S., Z. Chen, B. Liu. 2016. Mining aspect-specific opinion using a holistic lifelong topic model. Proceedings of the 25th International Conference on World Wide Web, pp. 167−176.

Wang, Y., H. Bai, M. Stanton, W. Chen, E. Chang. 2009. PLDA: Parallel Latent Dirichlet allocation for large-scale applications. *Lect. Notes Comput. Sci.* **5564**: 301–314.

Wang, T., Y. Cai, H. F. Leung, R. Y. K. Lau, Q. Li, H. Min. 2014. Product aspect extraction supervised with online domain knowledge. *Knowl.-Based Syst.* **74**: 86–100.

Weiss, A. 1984. ARMA models with ARCH errors. *J. Time Ser. Anal.* **5**(2): 129–143.

Williamson, S., A. Dubey, E. Xing. 2013. Parallel Markov Chain Monte Carlo for Nonparametric Mixture Models. The 30th International Conference on Machine Learning, pp. 98−106.

Wong, W. K., Z. X. Guo. 2010. A hybrid intelligent model for medium-term sales forecasting in fashion retail supply chains using extreme learning machine and harmony search algorithm. *Int. J. Prod. Econ.* **128**(2): 614–624.

Wu, Y., Q. Zhang, X. Huang, L. Wu. 2011. Structural opinion mining for graph-based sentiment representation. Proceedings of the 2011 Conference on EMNLP, pp. 1332–1341.

Xu, X., S. Tan, Y. Liu, X. Cheng, Z. Lin. 2012. Towards jointly extracting aspects and aspect-specific sentiment knowledge. Proceedings of the 21st ACM CIKM Conference, pp. 1895–1899.

Yang, B., C. Cardie. 2013. Joint inference for fine-grained opinion extraction. Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL'13), pp. 1640–1649.

Yu, X., Y. Liu, X. Huang, A. An. 2012. Mining online reviews for predicting sales performance: A case study in the movie domain. *IEEE Transac. KDE* **24**(4): 720–734.

Zadeh, L. 1965. Fuzzy sets. *J. Inf. Cont.* **8**: 338–353.

Zhang, W., R. Y. K. Lau, C. Li. 2014. Adaptive big data analytics for deceptive review detection in online social media. Proceedings of the Thirtieth-Fifth International Conference on Information Systems.

Zhao, W., J. Jiang, H. Yan, X. Li. 2010. Jointly modeling aspects and opinions with a maxent-lda hybrid. Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 56–65.

Zhou, M., L. Carin. 2012. Augment-and-Conquer negative binomial processes. Advances in Neural Information Processing Systems, pp. 2546−2554.

Supporting Information

Additional supporting information may be found online in the supporting information tab for this article:

Appendix S1: The Difference between the Classical HDP Model and the Proposed GP-HDP Model.