Cloud Computing for Extracting Price Knowledge from Big Data

George Suciu, Ciprian Dobre, Victor Suciu, Gyorgy Todoran, Alexandru Vulpe
University POLITEHNICA of Bucharest
Bd. Iuliu Maniu 1-3, Bucharest, Romania
george@beia.ro

Anca APOSTU
Academy of Economic Studies
Bucharest, Romania
ancaiapostu@gmail.com

***Abstract*—Customer price knowledge has been the object of considerable research in the past decades since the advent of online shopping. Furthermore, customers express online their personal opinions regarding the products or services they purchase, this activity becoming a habit for many people nowadays. However, due to the difficulty of analyzing such large datasets, extracting price knowledge from big data presents unique systems engineering and architectural challenges. The purpose of this paper is to analyze several existing solutions used for search and analysis of large volumes of data, with applicability in the retail field, and to present the results for price knowledge extraction from Big Data using Exalead CloudView technology. The main contribution of this paper consists in the development of several connectors and a data model based on properties and patterns specific for price calculations.**

Keywords: cloud computing; big data; price knowledge, cloud computing.

I. INTRODUCTION

The price setting process represents one of the key processes in a company and the aim of this process is to provide the mechanism to translate the longer term strategy of the company in terms of price positioning, market share goals, and product or service differentiation into the prices which are set or changed on a day-to-day basis within the competitive environment of the firm. The key elements for short-term pricing decision making within the longer term strategic constraints of price positioning, volume goals, and product and service differentiation include costs, sales volume and its variation with the price, the impact of competitor price, and interaction between prices of certain products within the company's product portfolio.

Customer price knowledge has been the object of considerable research in the past decades [1]. Authors of reference paper [2] cite over sixteen previous studies, most of which focus on measuring customers' short-term price knowledge of consumer packaged goods. In a typical study, customers are interviewed either at the point-of-purchase or in their home and asked to recall the price of a product, or alternatively, to recall the price they last paid for an item. In perhaps the most frequently cited study [3], asked supermarket shoppers to recall the price of an item shortly after they placed it into their shopping cart. Surprisingly, fewer than 50% of consumers accurately recall the price. Thus, despite the immediate recency of the purchase decision there is no improvement in the accuracy of the responses.

In another paper [4], combine survey data and a field experiment to investigate this prediction. In their study, they survey 14 customers and collect price recall measures for approximately two hundred products. They then conduct a field experiment in which they randomly assign the same items to one of three conditions. In the control condition, items are offered at the regular retail price. In the price cue condition, a shelf tag with the words "LOW prices" is used on an item. In the discount condition, the price is offered at a 12% discount from the regular price.

The authors show that both price cues and price discounts increase demand. But, consistent with theoretical predictions, the authors find that price cues are more effective on products for which customers have poor price knowledge. In contrast, price discounts are more effective when customers have better price knowledge. Together these results highlight the importance that price knowledge serves in determining the effectiveness of price changes and price cues.

Another paper [5] examines consumer price knowledge by comparing the actual market prices and consumer price estimates in the Finnish grocery market. Although the individual price estimates of consumers were found to differ significantly from the actual market prices, the medians of consumer price estimates and market prices were very close to each other for most of the products in our data. The study indicates that consumer price knowledge is not as poor as previously suggested by the results of point-of-purchase studies. The authors suggest that at least part of the weakness in consumer price knowledge can be explained by differences in market price variation.

The paper is organized as follows: Section II presents the rationale for Big Data in pricing, while Section III describes the methods and results for price knowledge extraction from Big Data. Finally, Section IV concludes the paper.

II. Rationale for Big Data in Pricing

The customers usually express their personal opinions regarding the products and services they purchase, this activity becoming a habit for many people nowadays. The online communities' continuous development, such as websites, blogs and social networks facilitate the exchange of information for the benefit of a growing number of users, increasing the social ties.

On a growing market, the companies are forced to follow closely the needs and requests of consumers, but also their suggestions, in order to develop a more clear vision regarding the quality of products and services. The customer satisfaction has become a significant factor which affects the size and market segment profitability. In response to this problem, companies are forced to regularly survey the satisfaction levels of customers, wanting to focus on feedback from them and effecting improvements in work practices and processes used in the company.

Influencing consumers depends not so much the brand (although in the case of this tradition is a very important), but depends heavily on user reviews, opinions left by them on various specialized websites. Life cycle of a product on the market is influenced by the information about this product coming from consumers. This information can be a process because in a first phase, after the product is launched user reviews will appear, and these are always changing. Following these opinions, the most trusted and secure analyses are selected [6].

Big data refers to the process of collecting and processing of very large data sets and to the associated systems and algorithms used to analyze these massive data sets. Architectures for big data usually range across multiple machines and clusters, and they commonly consist of multiple special purpose sub-systems. Coupled with the knowledge discovery process, big data movement offers many unique opportunities for organizations to benefit (with respect to new insights, business optimizations, etc.). Due to the difficulty of analyzing such large datasets, big data presents unique systems engineering and architectural challenges.

Currently, on the market there are many solutions for the search and analysis of large volumes of information solutions which are usually focused on semantic technologies for aggregating and collating both structured and unstructured data on private or public cloud platforms [7]. Besides the well-known Google, there have been developed and are under development solutions for Enterprise Business Applications such as CRM (Customer Relationship Management), ERP (Enterprise Resource Planning) and BI (Business Intelligence) and web applications, such as applications B2B (Business to Business), B2C (Business to Customer), using data from various sources (databases, web content, user generated content, etc.).

III. Results and Discussion for Price Knowledge Extraction from Big Data

In this section we provide results from using Exalead CloudView [8] for extracting price knowledge from Big Data. CloudView is a search platform which offers wide access to information on the infrastructure level and it is used for Search Based Applications (SBAs) for both online and enterprise level. The application combines several semantic technologies for the applications development, but also analysis technologies (qualitative and quantitative), used on the data presentation level, aiming to provide the right information to the user.

CloudView represents a tool that combines search technologies with Business Intelligence, and it is a platform for the Exalead search engine, which was designed to implement semantic processing and selective navigation for the data volumes from online environments, facilitating processes such as searching and analysis, and also enabling organizations to improve their knowledge and resources exploitation.

Basically, CloudView represents an instrument that allows the exploitation of huge data volumes, both structured and unstructured, including analytical databases [9]. Several connectors are already developed for these data sources, making the extracted price knowledge available for presentation in an intuitive search interface. There is also used a modular index that combines structures, terminologies and semantic technology platform formats, providing a continuous access and unceasingly use of resources, through server distribution techniques and data redundancy [10]. The manner in which Exalead CloudView processes and offers access to information is illustrated in Fig. 1.

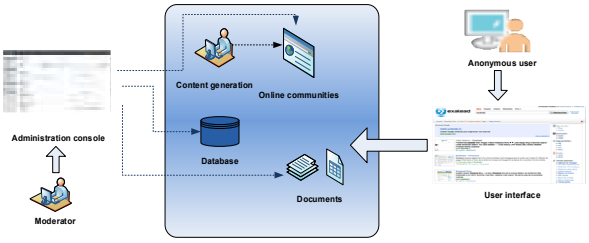

Fig. 1. Processing and accessing Big Data via Exalead CloudView

CloudView accesses data from different sources using the so-called connectors. Each connector uses the data source protocol to connect its own information source and access the documents which will indexed. Through connectors, there can be indexed several data types by adding new documents, updating or deleting existing documents, extracting current list of indexed documents and managing data security.

CloudView provides multiple interfaces for data management and application configuration itself. These interfaces are represented by several APIs (Application Programming Interface), including a Push API for creating custom connectors, a Search API used for the development of other applications, Search and Management API for configuring and managing the indexing and search processes. Push API allows the indexing of any type of data, coming from any source, and supports basic operations required for the development of new connectors, both managed and unmanaged. A managed connector is part of the code that runs in CloudView, code that can be developed in Java. An unmanaged connector can be developed in any language,

through CloudView's APIS, either Push APIs (in Java, C # and PHP) or through HTTP API.

Fig. 2 provides a simplified view of the indexing process. The functionality can be described as follows: the connectors access the data sources and convert files into documents; these documents are sent to the Push Server and further are divided into groups, in the Analysis phase.

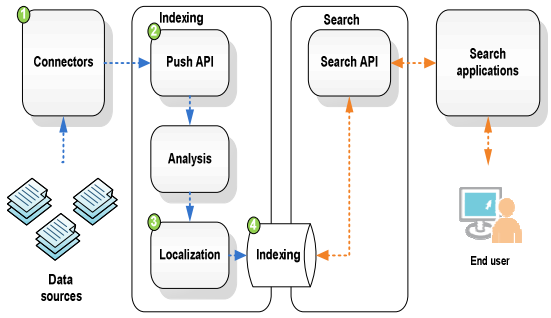

Fig. 2. Simplified view of the indexing process

The analysis is performed sequentially, each document being processed, making retrieval of text, semantic processing, and other custom operations and location. The analysis also calculates the data to update the index. Once indexing is done and updated, the new documents are available for searching.

Connectors operate early in the CloudView indexing process, sending documents from different sources to a Push API server through the PAPI protocol. If the connector server is already existing, all new connectors will be automatically associated with it.

A. Configuration Methodology for CloudView

The CloudView configuration methodology is based on the connector's development, especially in the indexing process. This methodology is performed within the Administration Console, where the user can perform various configurations for the connectors, such as deleting all documents and restoring the connector's state, synchronizing indexed documents, or stopping the synchronization update for a specific type of connector (procedure that does not delete documents already processed). Standard connectors of CloudView include files, databases, HTTP, XML etc.

File type connectors are used for crawling local file systems or remote systems that are shared within the network. The search path configuration depends on the operating system on which CloudView runs In order to create a file type connector, it is necessary to ensure for the search system the access rights for the file system, i.e. file system rights to access the user account that is configured on CloudView. There also can be used regular expressions in order to specify a certain path (select regexp).

The JDBC database connector allows indexing of SQL database content, by using configurable queries for extracting data records. These records are then processed to compose the documents to be indexed. Documents are built by selecting the columns in the list returned by the synchronization queries. For indexing the database, a connector can be set to do a full synchronization, which indexes the entire database or set to one of the incremental modes for indexing only a part of the database. Incremental modes can be used when the table contains a timestamp representing the time of insertion or a unique serial number. When using incremental synchronizing, the connector takes the timestamp or the serial number from the column and assigns it to a variable in the query to retrieve new rows in the database.

HTTP connectors can load any URL address, provided that it is accessible from the server where CloudView is installed. The URL is taken, transmitted sequentially and then the links are extracted from that sequence. The main components of the CloudView system are the Crawler Server, Crawler Manager and Push Server. All other connectors, except those, that are hosted on the HTTP Server Connector, are installed on the Crawler Server. In the URL field, the site address is entered and some rules for indexing are set, for example reloading behaviour. Smart Refresh reloads URLs in the list and adds them to the queue, except for the recent uploads. URLs added by the reloading loop have a lower priority than newly discovered ones, so the reload does not slow the discovery of new pages.

B. The Data Model for the Price Optimization

The development of any search application in Exalead CloudView requires defining what data to include in the index schema, and then configuring one or more search logics to control how the documents are presented to users as search results. Index fields are used for searching and to display data in hit content in the results that are returned.

Categories store static facet values. These values display in the Refinements panel of the search results as well as in hit content. Static facets allow users to narrow their search results by focusing on a certain aspect of the results, such as a particular country or product line, price value within a range etc. Furthermore, by using big data mining it is possible to identify patterns between indicators and predict price variations [11].

The data model represents a starting point for configuring the index and the search processes. Through a data model, it is allowed to manage different types of data according their purpose and their nature.

There can be created a set of property types like alphanumeric, numeric, geographic. After scanning the data sources, the settings which have been defined for the properties generate new index fields and categories inside the index schema and also several meta elements or prefix handlers according to the data logic.

When a configuration is applied, the high-level view provided by the Data Model is expanded into the multiple index and search elements. A common challenge when creating properties is to know which metas belong to the corpus. There are several ways in which the available metas can be explored using the CloudView data model.

When creating a data model property, it can be chosen one of the following field types: either an index field or a category

facet only, or both. The most common indexing method is to assign a dedicated Indexfield attribute in XML configuration to the property, represented in Fig. 3. The property can then be made searchable (user queries can be applied to this field) and retrievable (the field can display in the search results).

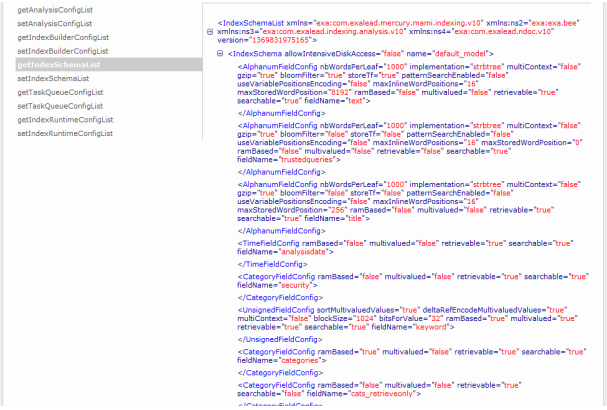

Fig. 3. XML configuration of the property

The property can also be stored as a facet and a category will be created with the value of the property. Faceted navigation will automatically be created for this property. In that case, it is possible to search for the value of the property. If the property belongs to the default class, the property can be searched with a prefix format of property:value field. Otherwise, it can be searched with a prefix format of classname_property:value.

The property can be made searchable without prefix, which means in addition to the indexing determined by other options, the property is indexed as searchable in the field called text. This allows users to search this property without specifying a prefix in the query. When storing a numerical property as a facet, only equality search is possible. Range search is only possible in a dedicated index field. On numerical properties with a dedicated index field, searchable with prefix includes searching using the range operators.

While the data model simplifies the set up of the most common features for index fields and category facets, sometimes the user may need to customize indexing or search behavior. Furthermore, it is possible to customize elements generated by data model expansion. For example, the user can modify the default clipping options for the data model-generated facets on a facet-by-facet basis.

IV. Conclusions

In this paper we analyzed several existing solutions used for search and analysis of large volumes of data, with applicability in the retail field. Assessment of prices and product reviews can be developed using a search engine, by using semantic processing, which provides relevant results to the search query in case of incomplete or inaccurate results.

In order to create a model for the price evaluation, we developed several connectors and a data model based on properties and patterns which are likely to provide innovative solutions to existing problems. An evaluation can be based on classification opinions and calculation of notes, depending on the positive or negative reviews of their number, etc. As future work we envision accessing this information either online or by importing and compiling various formats (database, XML file).

Acknowledgment

This work has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/159/1.5/S/134398 and was supported in part by the projects no. 305E/2011 ”Cloud-based Automation of ERP and CRM software for Small Businesses – Cloud Consulting”, grant agreement no. 643963 ”SWITCH” and no. 319E/2012 ”Network Management System Development and Monitoring – NMSDMON”.

References

[1] A. Apostu, “Price conditions and price calculation in nowadays retail systems. Price knowledge extracted from Big Data,” Emerging Markets Queries in Finance and Business, 2013, pp. 20-27.

[2] K. B. Monroe, and Y. L. Angela, “Remembering versus Knowing: Issues in Buyers’ Processing of Price Information,” Journal of the Academy of Marketing Science, Vol. 27, No. 2, 1999, pp. 207-225.

[3] P. R. Dickson, and A. G. Sawyer, ”The price knowledge and search of supermarket shoppers,” The Journal of Marketing, 1990, pp. 42-53.

[4] E. T. Anderson, K. C. Edward, H. Bari, and I. S. Duncan, Using Price Cues, MIT, Cambridge MA, 2007.

[5] V. Aalto-Setälä, and A. Raijas, “Actual market prices and consumer price knowledge,” Journal of Product & Brand Management, Vol. 12, No. 3, 2003, pp. 180-192.

[6] R. Eckstein, Interactive Search Processes in Complex Work Situations-A Retrieval Framework, Vol. 10, 2011, University of Bamberg Press.

[7] G. Suciu, E. G. Ularu, and R. Craciunescu,”Public versus private cloud adoption—A case study based on open source cloud platforms,” 20th Telecommunications Forum (TELFOR), IEEE, 2012, pp. 494-497.

[8] G. Grefenstette, and L. Wilber, "Search-based Applications: At the Confluence of Search and Database Technologies," Synthesis Lectures on Information Concepts, Retrieval, and Services 2.1, 2010, pp. 1-141.

[9] E.G. Ularu, F.C. Puican, A. Apostu, and G. Suciu, ”Analytical Databases for the Cloud and Virtualization,” 12th International Conference on Informatics in Economy (IE), 2013, pp. 337-341.

[10] T. Seymour, D. Frantsvog, and S. Kumar S. "History of search engines," International Journal of Management & Information Systems (IJMIS), vol. 15, no. 4, 2011, pp. 47-58.

[11] L. Lima, F. Portela, M. F. Santos, A. Abelha, and J. Machado, "Big Data for Stock Market by Means of Mining Techniques," In New Contributions in Information Systems and Technologies, Springer International Publishing, 2015, pp. 679-688.