iMiner: Mining Inventory Data for Intelligent Management

Lei Li[†], Chao Shen[†], Long Wang[†], Li Zheng[†], Yexi Jiang[†], Liang Tang[†], Hongtai Li[†]
Longhui Zhang[†], Chunqiu Zeng[†], Tao Li[†], Jun Tang[‡], Dong Liu[‡]

[†]School of Computing and Information Sciences
Florida International University
Miami, FL 33199
{lli003,taoli}@cs.fiu.edu

[‡]ChangHong Hongxin Software Co., Ltd
35 East Mianxing High-Tech Park
Mianyang, Sichuan, China 621000
{tang.jun,liu.dong}@changhong.com

ABSTRACT

Inventory management refers to tracing inventory levels, orders and sales of a retailing business. In the current retailing market, a tremendous amount of data regarding stocked goods (items) in an inventory will be generated everyday. Due to the increasing volume of transaction data and the correlated relations of items, it is often a non-trivial task to efficiently and effectively manage stocked goods. In this demo, we present an intelligent system, called iMiner, to ease the management of enormous inventory data. We utilize distributed computing resources to process the huge volume of inventory data, and incorporate the latest advances of data mining technologies into the system to perform the tasks of inventory management, e.g., forecasting inventory, detecting abnormal items, and analyzing inventory aging. Since 2014, iMiner has been deployed as the major inventory management platform of ChangHong Electric Co., Ltd, one of the world's largest TV selling companies in China.

Categories and Subject Descriptors: H.2.8[Database Applications]: Data Mining

Keywords: Inventory Management; Inventory Forecasting; Anomaly Detection; Inventory Aging

1. INTRODUCTION

Inventory management is often an indispensable process for most retailing companies, as it has the functionalities to avoid product overstock and outages. It refers to the general process of monitoring the fluctuant flow of goods (items) into and out of an existing inventory. This process usually generates two types of time series data, representing the amount of stock in/out evolving over time. Complete inventory management also seeks to control the costs associated with inventory operations [10].

In current retailing businesses, a huge amount of items are often stored at different locations of a supply network to precede the planned course of sales. For example, in our demo case, the retailing vendor has 251,874 items in total, with 132,140 transactions daily on an average. The increasing volume of such data renders it difficult to manually manage the inventory, e.g., to specify the shape and percentage of stocked items. In addition, some items may have correlations with others. For example, when a customer is buying a TV, he/she may also choose multiple auxiliary products, e.g., TV mount or DVD players. Such correlations further increase the difficulty of efficient inventory management. Existing inventory management softwares, such as inFlow[1] and Inventoria[2], provide functionalities to facilitate businesses to manage the inventory. However, most of these systems only rely on statistical analysis of the existing inventory data, and have very limited capability of intelligent management, e.g., forecasting item demand and detecting abnormal patterns of item inventory transactions.

To address the limitations of existing systems, we design and develop iMiner to assist retailing businesses in efficiently performing inventory management. iMiner provides a set of key functionalities that help businesses conveniently and effectively manage huge volume of inventory data. Specifically, the system has the following merits:

- *Efficient support of large-scale inventory data analysis.* The analytical platform is built on a distributed system (FIU-Miner [9]) to support high-performance data analysis. The platform manages all the transaction data in a distributed environment, which is capable of configuring and executing data preprocessing and data analysis tasks in an automatic way.
- *Effective management of complex analysis tasks.* iMiner integrates appropriate data mining algorithms and adapts them to the problem of analyzing inventory data. In particular, the system 1) adopts various regression models and combines them with time series data analysis to fulfill the task of *inventory forecasting*; 2) employs context-aware anomaly detection algorithms to *identify abnormal items*; and 3) utilizes statistical regression models to perform *inventory aging analysis*.

2. SYSTEM OVERVIEW

iMiner has been designed and developed by a research team consisting of 12 members for one year. The system is designed under the following principles: 1) The methods used in the system should provide accurate, and more importantly, interpretable results; 2) The system should provide users with interactive functionalities; and 3) The system should be able to handle large-scale data analysis. A demo

CIKM'14, November 3–7, 2014, Shanghai, China.
ACM 978-1-4503-2598-1/14/11.
http://dx.doi.org/10.1145/2661829.2661848.

[1] http://www.inflowinventory.com.

[2] http://www.nchsoftware.com/inventory.

can be found in http://matrix.cs.fiu.edu/iMiner. Figure 1 presents an overview of the system architecture of `iMiner`. The system is composed of four layers: *User Interface*, *Data Analysis Layer*, *Task and System Management Layer*, and *Physical Resource Layer*.

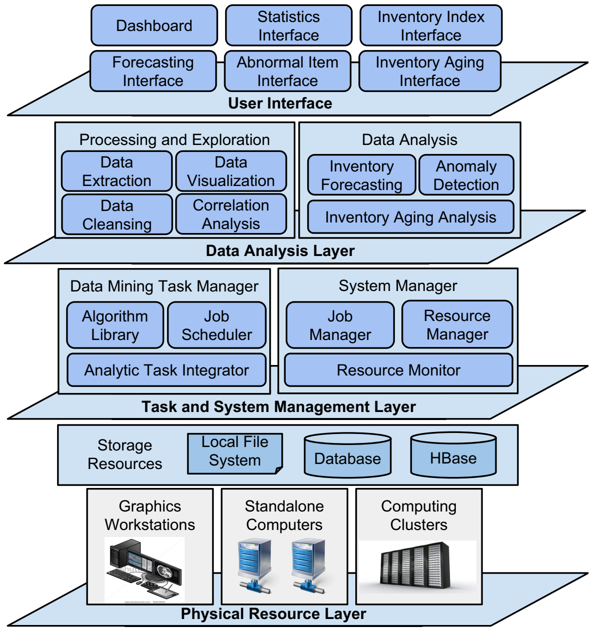


Figure 1: An overview of the system architecture.

Existing inventory management softwares have two limitations when applied to the practice: 1) They do not support handy algorithm plug-in; and 2) They do not support large scale analysis tasks running in parallel in heterogeneous environments. *Task and System Management Layer* provides a fast, integrated, and user-friendly system to address the aforementioned limitations, where all the data analysis tasks in *Data Analysis Layer* can be configured as workflows and automatically scheduled. This system is built on our previous large-scale data mining system, FIU-Miner. More details of this system can be found in [9].

Data Analysis Layer consists of appropriate data mining solutions to the corresponding tasks of inventory management, including *Inventory Forecasting*, *Anomaly Detection*, and *Inventory Aging Analysis*. In Section 3, more details are provided by presenting our data mining solutions customized for inventory management tasks. The system also provides basic data processing and exploration functionalities.

User Interface contains various interactive interfaces for inventory operations. Specifically, it provides *Dashboard* and *Statistics Interface* to allow users to have an overview of the current inventory status. In addition, several key indices of inventory, e.g., turnover rate and stock-to-use ratio, are presented in *Inventory Index Interface*, assisting users in promptly querying the status of a particular item.

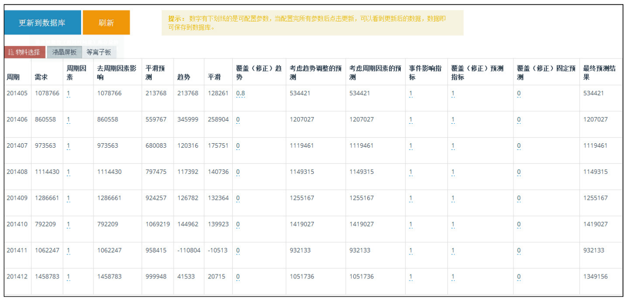
3. FUNCTIONAL COMPONENTS

3.1 Inventory Forecasting

The primary goal of inventory forecasting is to minimize the inventory loading. Excess inventory is sub-optimal as it requires additional maintenance and cost. Hence, in inventory management, it is imperative to perform accurate forecasting so as to reduce the inventory investment and risk of obsolescence. A common practice of inventory forecasting is to predict the demand of a specific item in the future, and reserve the amount of item based on the forecasting result. This often refers to as demand forecasting. In existing solutions to this problem, this is often achieved by calculating past averages. However, the demand of an item is determined by not only contemporary comparison, but also a list of factors, such as seasonality, trend, and special events. Hence, a much more accurate and optimal forecasting method is expected.

To this end, we design and implement a forecasting mechanism in `iMiner` based on the past inventory transaction data. It takes as input the past transaction data of an item (as a real-valued time series), and processes the data into a list of instances. In each instance i, the amount of stock out of the item in timestamp i is treated as the label, and the values of stock out in i's previous k timestamps are regarded as the features. We then utilize various regression algorithms, such as Neural Network, Linear Regression, SVM [2], Gaussian Process [7], regression tree [1], and gradient descent regression tree [3], to build regression models with optimized parameters for each item. These models are updated in a daily basis. An ensemble method [8] is employed to aggregate the predictions of these models.

The obtained result from the ensemble is purely originated from the perspective of the data itself, without considering the characteristics of inventory forecasting. To enhance the interpretability of forecasting, we treat the ensemble result as the forecasting basis, and further propose a dynamic model that takes into account various factors of inventory data, including seasonality, trend, and special events. Seasonality refers to the portion of item demand fluctuation accounted for by a reoccurring pattern [6]. In inventory management, such patterns often repeat over time. Comparatively, trend is the portion of item demand without reoccurrence. For instance, a trend may show a period of growth followed by a leveling off. Special events, such as holidays and sales promotion, may have great impact on the demand of items. These factors are integrated into an interactive interface shown in Figure 2 to provide dynamic prediction for demand forecasting.



提示：数字有下划线的是可配置参数，当配置完所有参数后点击更新，可以看到更新后的数据，数据即可保存到数据库。

周期	需求	周期因素	去周期因素影响	平滑预测	趋势	平滑	覆盖（修正）趋势	考虑趋势调整的预测	考虑周期因素的预测	事件影响指标	覆盖（修正）预测指标	覆盖（修正）固定预测	最终预测结果
201405	1078766	1	1078766	213768	213768	128261	0.8	534421	534421	1	1	0	534421
201406	860558	1	860558	559767	345999	258904	0	1207027	1207027	1	1	0	1207027
201407	973563	1	973563	680083	120316	175751	0	1119461	1119461	1	1	0	1119461
201408	1114430	1	1114430	797475	117392	140736	0	1149315	1149315	1	1	0	1149315
201409	1286661	1	1286661	924257	126782	132364	0	1255167	1255167	1	1	0	1255167
201410	792209	1	792209	1069219	144962	139923	0	1419027	1419027	1	1	0	1419027
201411	1062247	1	1062247	958415	-110804	-10513	0	932133	932133	1	1	0	932133
201412	1458783	1	1458783	999948	41533	20715	0	1051736	1051736	1	1	0	1349156

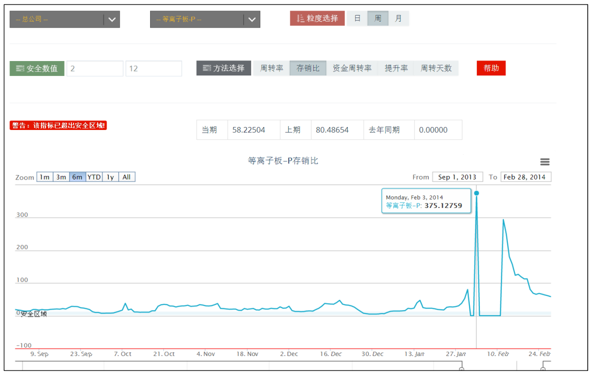
Figure 2: Dynamic forecasting of item demand.

3.2 Inventory Anomaly Detection

Monitoring inventory index for anomaly detection is a very important task in inventory management. This problem becomes further difficult in the Big Data era as the data scale increases dramatically and the type of anomalies gets more complicated.

In our system, in addition to providing traditional statistical methods (e.g., parametric and non-parametric methods) and proximity-based methods [5], we also design and develop a context-aware anomaly detection method to identify abnormal items in inventory data. The data in inventory management includes both transaction data (i.e., time series) and item-related data (i.e., item catalog and description). We combine transaction data and product-related data and identify context clusters as items having similar transaction behaviors with similar item attributes. An instance (i.e., an item in a time interval) is anomalous if it is far away from any context clusters. To identify context clusters, we make use of spectral clustering where the similarity between transaction data (measured by their principal components) and the similarity between the item-related data (measured by their textual attributes) are combined with a learned kernel [4].

Our system builds the unsupervised model of context clusters based on a training set and then applies the model for new test instances. New test instances arriving every day can be cross-checked with the learned model and the model can also be updated with the new data. Figure 3 shows an example of the anomaly detection interface.



安全数值 2 12

警告：该指标已超出安全区域!

当期	58.22504	上期	80.48654	去年同期	0.00000

等离子板–P存销比

From Sep 1, 2013 To Feb 28, 2014

Monday, Feb 3, 2014
等离子板–P: 375.12759

Figure 3: An example of anomaly detection.

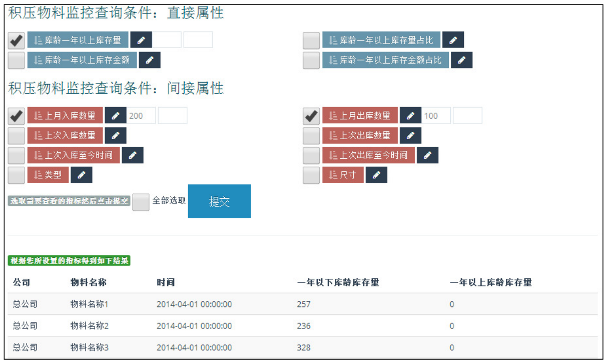
3.3 Inventory Aging Analysis

The main purpose to monitor and analyze inventory aging is to prevent items from overstocking and reduce the overstocked items. In our system, an overstocked item at the time t is an item with the amount more than $x\%$ (e.g., 30%) over y (e.g., 6 months) old, where x and y can be set by users. We provide in the system both basic tools and advanced tools to analyze the inventory aging.

The basic tools allow users to visualize inventory aging distributions of a given item and compare current and historical inventory aging changes among different items. The advanced tools are able to help users find attributes of items correlated with overstocking. This further allows users to monitor the related attributes, and especially pay attention to those items of which the value of the related attribute is above or below a predefined threshold.

To indicate how related the item attributes are with overstocking, we model it as a feature selection problem. For each item and each timestamp when the item exists in stock, we generate a data sample, (x_i, y_i), where x_i are the attributes of the item and y_i is the label with value 1 indicating the item at that time is overstocked, and value 0 otherwise. The label information can divide the entire item collection $\mathcal{X}$ to overstocked items $\mathcal{X}_1$ and non-overstocked items $\mathcal{X}_0$. The system integrates several feature selections methods including Information Gain, mRMR and ReliefF to rank the attributes, and allows users to configure which one will be used at running time. For each attribute, users can further compare its histograms on $\mathcal{X}_1$ and $\mathcal{X}_0$ to obtain a more intuitive sense of the relationship between the attribute and overstocking.

Using the related attributes, users can impose queries to retrieve the items that are likely to overstock and put them into the monitoring list. Figure 4 shows the interface of querying items using multiple attributes. A query is a logical conjunction of a set of literals, each of which corresponds to an attribute. For a continuous attribute a, its literal is in the form of $L_a \leq a \leq H_a$, where L_a and H_a are the lower and upper threshold of the attribute a, respectively. For a discrete attribute, the condition is in the form of $a = V_a$, where V_a is the given value of a. Given a literal for an attribute a, we can retrieve a subset of the items $\mathcal{X}^a \subset \mathcal{X}$. To help users compose the query, for attribute a, the system recommends a literal as $[L_a^*, +\infty]$ or $[-\infty, H_a^*]$ for continuous attributes and V_a^* for discrete attributes, with the goal of maximizing the probability of overstocking in the corresponding subset $p(y_i = 1|\mathcal{X}_a)$.



积压物料监控查询条件：直接属性

☑ 库龄一年以上库存量
□ 库龄一年以上库存金额
□ 库龄一年以上库存量占比
□ 库龄一年以上库存金额占比

积压物料监控查询条件：间接属性

☑ 上月入库数量 200
□ 上次入库数量
□ 上次入库至今时间
□ 类型
☑ 上月出库数量 100
□ 上次出库数量
□ 上次出库至今时间
□ 尺寸

选取需要查看的指标然后点击提交 □ 全部选取

根据您所设置的指标得到如下结果

公司	物料名称	时间	一年以下库龄库存里	一年以上库龄库存里
总公司	物料名称1	2014-04-01 00:00:00	257	0
总公司	物料名称2	2014-04-01 00:00:00	236	0
总公司	物料名称3	2014-04-01 00:00:00	328	0

Figure 4: Attributes in inventory aging analysis.

ACKNOWLEDGMENT

The work was supported in part by the National Science Foundation under grants DBI-0850203, CNS-1126619, and IIS-1213026, the Army Research Office under grants W911NF-10-1-0366 and W911NF-12-1-0431, and an FIU Dissertation Year Fellowship.

4. REFERENCES

[1] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. 1984.

[2] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *TIST*, 2(3):27, 2011.

[3] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

[4] G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *JMLR*, 5:27–72, 2004.

[5] T. Pang-Ning, M. Steinbach, V. Kumar, et al. *Introduction to data mining*. 2006.

[6] R. S. Pindyck and D. L. Rubinfeld. *Econometric models and economic forecasts*, volume 4. 1998.

[7] C. E. Rasmussen. Gaussian processes for machine learning. 2006.

[8] D. A. Unger, H. van den Dool, E. O'Lenic, and D. Collins. Ensemble regression. *Monthly Weather Review*, 137(7):2365–2379, 2009.

[9] C. Zeng et al. Fiu-miner: a fast, integrated, and user-friendly system for data mining in distributed environment. In *SIGKDD*, pages 1506–1509. ACM, 2013.

[10] P. H. Zipkin. *Foundations of inventory management*, volume 2. 2000.