

# Mining Retail Transaction Data for Targeting Customers with Headroom - A Case Study

Madhu Shashanka and Michael Giering

**Abstract** We outline a method to model customer behavior from retail transaction data. In particular, we focus on the problem of recommending relevant products to consumers. Addressing this problem of filling *holes in the baskets* of consumers is a fundamental aspect for the success of targeted promotion programs. Another important aspect is the identification of customers who are most likely to spend significantly and whose potential spending ability is not being fully realized. We discuss how to identify such customers with *headroom* and describe how relevant product categories can be recommended. The data consisted of individual transactions collected over a span of 16 months from a leading retail chain. The method is based on Singular Value Decomposition and can generate significant value for retailers.

## 1 Introduction

Recommender systems have recently gained a lot of attention both in industry and academia. In this paper, we focus on the applications and utility of recommender systems for brick-and-mortar retailers. We address the problem of identifying shoppers with high potential spending ability and presenting them with relevant offers/promotions that they would most likely participate in. The key to successfully answering this question is a system that, based on a shopper's historical spending behavior and shopping behaviors of others who have a similar shopping profile, can predict the product categories and amounts that the shopper would spend in the future. We present a case study of a project that we completed for a large retail chain. The goal of the project was to mine the transaction data to understand shopping behavior and target customers who exhibit *headroom* - the unmet spending potential of a shopper in a given retailer.

The paper is organized as follows. Section 2 presents an overview of the project and Section 3 provides a mathematical formulation of the problem. After presenting

---

Please use the following format when citing this chapter:

Shashanka, M. and Giering, M., 2009, in IFIP International Federation for Information Processing, Volume 296; *Artificial Intelligence Applications and Innovations III*; Eds. Iliadis, L., Vlahavas, I., Bramer, M.; (Boston: Springer), pp. 347–355.

a brief technical background in Section 4, we present details of the methodology and implementation in Section 5. We end the paper with conclusions in Section 6.

## 2 Project Overview - Dataset and Project Goals

Data from every transaction from over 350 stores of a large retail chain gathered over a period of 16 months was provided to us. Data is restricted to transactions of regular shoppers who used a “loyalty card” that could track them across multiple purchases. For every transaction completed at the checkout, we had the following information: date and time of sale, the receipt number (*ticket number*), loyalty-card number of the shopper (*shopper number*), the product purchased (given by the *product number*), product quantity and the total amount, and the store (identified by the *store number*) where the transaction took place. A single shopping trip by a customer at a particular store would correspond to several records with the same shopper number, the same store number, and the same ticket number, with each record corresponding to a different product in the shopping cart. There were 1,888,814 distinct shoppers who shopped in all the stores in the period of data collection.

Along with the transaction data, we were also given a *Product Description Hierarchy* (PDH). The PDH is a tree structure with 7 distinct levels. At level 7, each leaf corresponds to an individual product item. Level 0 corresponds to the root-node containing all 296,387 items. The number of categories at the intermediate levels, 1 through 6, were 9, 50, 277, 1137, 3074 and 7528 respectively. All analysis referred to in this paper was performed at level 3 (denoted as  $L_3$  henceforth).

There were two main aspects in the project. The first was to identify those shoppers who were not spending enough to reflect their *spending potential*. These could be shoppers who have significant disposable income and who could be persuaded to spend more or regular shoppers who use the retail chain to fulfill only a part of their shopping needs. In both cases, the customers have *headroom*, i.e. unrealized spending potential.

Once headroom customers have been identified, the next logical step is to find out product categories that would most likely interest them and to target promotions at this group. This is the problem of filling *holes in the baskets*, by motivating them to buy additional products that they are not currently shopping for. In many respects this is similar to a movie recommender problem, where instead of movie watching history and movie ratings of each person, we have the shopping history and spends.

## 3 Mathematical Formulation

In this section, we introduce mathematical notation and formulate the problem. Let  $S_{cpm}$  denote the amount spent by shopper  $c$  in the product category  $p$  during month  $m$ , and  $n_{cpm}$  denote the number of items bought in that product category. For the

purposes of clarity and simplicity, let us denote the indices  $\{c, p, m\}$  by the variable  $\tau$ . In other words, each different value taken by  $\tau$  corresponds to a different value of the triplet  $\{c, p, m\}$ . Let us define the quantity *Spend Per Item* (SPI) as  $I_\tau = (S_\tau/n_\tau)$ . The above quantities can be represented as 3-dimensional matrices  $\mathbf{S}$ ,  $\mathbf{n}$  and  $\mathbf{I}$  respectively, where the three dimensions correspond to shoppers, product categories and months. These matrices are highly sparse with entries missing for those values of  $\tau = \{c, p, m\}$  that correspond to no data in the data set (i.e. items that were not bought by shoppers). Let  $\tau_0$  represent the set of values of  $\{c, p, m\}$  for which there is no data and let  $\tau_1$  represent the set of values of  $\{c, p, m\}$  for which data is present, i.e.  $\tau = \{\tau_0 \cup \tau_1\}$ .

The first problem is to estimate each shopping household’s unrealized spending potential in product categories that they haven’t bought. This information can then be used for targeting and promotions. Mathematically, the problem is to estimate  $\mathbf{S}_{\tau_0}$  given the values in  $\mathbf{S}_{\tau_1}$ .

The second problem is to identify a set of shoppers who have headroom. Although subjective, the most common usage of this term refers to customers who have additional spending potential or who are not using a retailer to fill shopping needs that could be met there. There are many possible proxy measures of headroom, each focusing on different aspects of shopping behavior. We chose to derive four of these headroom metrics<sup>1</sup> - (a) total actual spend, (b) total actual SPI, (c) residue between model spend and actual spend, and (d) frequency of shopping.

For ease of comparison between the metrics and for the purpose of consolidating them later, we express each metric for all the shoppers in probabilistic terms. Our first three metrics are well suited for representation as *standard z-scores*. The frequency metric requires a mapping to express it as a *standard z-score*. For every metric, we choose a value range and define shoppers with z-scores in this range as exhibiting headroom. Section 5 details how the scores are consolidated.

## 4 Background: Singular Value Decomposition

The Singular Value Decomposition (SVD) factorizes an  $M \times N$  matrix  $\mathbf{X}$  into two orthogonal matrices  $\mathbf{U}$ ,  $\mathbf{V}$  and a diagonal matrix  $\mathbf{S} = \text{diag}(\mathbf{s})$  such that

$$\mathbf{U}\mathbf{S}\mathbf{V}^T = \mathbf{X} \quad \text{and} \quad \mathbf{U}^T\mathbf{X}\mathbf{V} = \mathbf{S}. \tag{1}$$

The elements of  $\mathbf{s}$  are the *singular values* and the columns of  $\mathbf{U}$ ,  $\mathbf{V}$  are the left and right *singular vectors* respectively. The matrices are typically arranged such that the diagonal entries of  $\mathbf{S}$  are non-negative and in decreasing order. The  $M$ -dimensional columns of  $\mathbf{U}$ ,  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_M\}$ , form an orthonormal matrix and correspond to a linear basis for  $\mathbf{X}$ ’s columns (span the *column space* of  $\mathbf{X}$ ). Also, the  $N$ -dimensional

---

<sup>1</sup> These metrics can be measured separately for each product category if desired. We mention the overall metrics across categories for the sake of simplicity in exposition.

rows of  $\mathbf{V}^T$ ,  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N\}$ , form an orthonormal matrix and correspond to vectors that span the *row space* of  $\mathbf{X}$ .

Given a data matrix  $\mathbf{X}$ , a reduced rank SVD decomposition  $\mathbf{U}_{M \times R} \mathbf{S}_{R \times R} \mathbf{V}_{R \times N}^T = \mathbf{X}_{M \times N}$ , where  $R < \min(M, N)$  is an approximate reconstruction of the input. This *thin* SVD is also the best rank- $R$  approximation of  $\mathbf{X}$  in the least squares sense. The singular values are indicative of the the significance of the corresponding row/column singular vectors in reconstructing the data. The square of each singular value is proportional to the variance explained by each singular vector. This allows one to determine a rank for the desired decomposition to express predetermined percentage of the information (variance) of the data set for approximation. In practice, data is typically centered at the origin to remove centroid bias. In this case, SVD can be interpreted as a Gaussian covariance model.

### SVD for Recommendation

Several well-known recommender systems are based on SVD and the related method of Eigenvalue decomposition (eg. [6, 2]). Let the matrix  $\mathbf{X}$  represent a matrix of consumer spends over a given period. Each of the  $N$  columns represents a different shopper and each of the  $M$  rows represents a different product.  $X_{mn}$ , the  $mn$ -th entry in the matrix, represents how much shopper  $n$  spent on product  $m$ . Consider the  $k$ -rank SVD  $\mathbf{U}' \mathbf{S}' \mathbf{V}'^T \approx \mathbf{X}$ . The subspace spanned by the columns of  $\mathbf{U}'$  can be interpreted as the  $k$  most important types of “shopping profiles” and a location can be computed for each shopper in this *shopping profile space*. The relationship between  $\mathbf{x}_n$ , the  $n$ -th column of  $\mathbf{X}$  representing the spends of the  $n$ -th shopper, and his/her location in the shopping profile space given by a  $k$ -dimensional vector  $\mathbf{p}_n$  is given by  $\mathbf{p}_n = \mathbf{U}'^T \mathbf{x}_n$ . It is easy to show that  $\mathbf{p}_n$  is given by the  $n$ -th row of the matrix  $\mathbf{V}' \mathbf{S}'$ . This vector  $\mathbf{p}_n$  underlies all SVD-based recommender systems, the idea is to estimate  $\mathbf{p}_n$  and thus obtain imputed values for missing data in  $\mathbf{x}_n$ . This also enables one to identify shoppers with similar shopping profiles by measuring the distance between their locations in the *shopping profile space* [6, 7]. Similarly, the *product space* given by columns of  $\mathbf{V}'$  show how much each product is liked/disliked by shoppers belonging to the various shopping profiles. These subspaces are very useful for subsequent analysis such as clustering and visualization.

Despite its advantages, the main practical impediment to using a thin SVD with large data sets is the cost of computing it. Most standard implementations are based in Lanczos or Ritz-Raleigh iterations that do not scale well with large data sets. Such methods require multiple passes through the entire data set to converge. Several methods have been proposed to overcome this problem for fast and efficient SVD computations [3, 5]. In this paper, we use the iterative incremental SVD implementation (IISVD) [2, 1] which can handle large data sets with missing values. Details of the implementation are beyond the scope of this paper.

## 5 Methodology

### Data Preprocessing

For our retail sales data, the assumption of log-normal distribution of spend and spend per item on each  $L_3$  product category and for the overall data are very good. There are always issues of customers shopping across stores, customers buying for large communities and other anomalous sales points. We first eliminate these outliers from further analysis. We screen shoppers based on four variables - the total spend amount, the number of shopping trips, the total number of items bought, and the total number of distinct products bought. The log-distribution for each variable showed us that the distributions were close to normal, but containing significant outlier tails corresponding to roughly 5% of the data on either end of the distribution. All the shoppers who fall in the extreme 5% tails are eliminated. This process reduces the number of shoppers from 1,888,814 to 1,291,114.

The remaining data is log-normalized and centered. The relatively small divergence from normality at this point is acceptable for the justification of using the SVD method (which assumes Gaussian data) to model the data.

### Clustering

A key to accurate modeling of retail data sets of this size is the ability to break the data into subsegments with differing characteristics. Modeling each segment separately and aggregating the smaller models gives significant gains in accuracy. In previous work [4] we utilized demographic, firmographic and store layout information to aid in segmentation. In this project, we derived our segments solely from shopping behavior profiles.

Shopping behavior profiles are generated by expressing each shopper's cumulative spend for each  $L_3$  product category in percentage terms. The main reason for considering percentage spends is that it masks the effect of the shopper household size on the magnitude of spend and focuses on the relative spend in different  $L_3$  categories. For example, shoppers with large families spend more compared to a shopper who is single and lives alone. We believe that this approach produces information more useful for discriminating between consumer lifestyles.

We begin by creating a  $150 \times 1$  vector<sup>2</sup> of percent spends per  $L_3$  category. A dense matrix  $\mathbf{X}$  containing one column for each shopper is constructed. We generate the SVD decomposition,  $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ , using IISVD.

From experience we know that we can assume a noise content in retail data of more than 15 percent. Using this as a rough threshold, we keep only as many singular vectors whose cumulative variance measures sum to less than 85 percent of the overall data variance. In other words, the rank we choose for the approximation  $\mathbf{U}'\mathbf{S}'\mathbf{V}'^T$  is the minimum value of  $k$  such that  $(\sum_{i=1}^k s_i^2) / (\sum_{i=1}^{150} s_i^2) \geq 0.85$ . We

<sup>2</sup> We were asked to analyze only a subset of 150 from among the total 277 categories.

computed the rank to be 26. As we mentioned earlier in Section 4, the rows of matrix  $\mathbf{V}'\mathbf{S}'$  correspond to locations of shoppers in this 26-dimensional *shopping profile space*. We segment the shoppers by running K-means clustering on the rows of this matrix. By setting a minimum cluster distance, we limited the number of distinct clusters to 23. Characterizing each cluster to better understand the differentiating customer characteristics unique to each cluster was not carried out. This time intensive process can provide significant efficiencies and be valuable for projects designed to accommodate iterative feedback.

### Imputation of Missing Values

Consider  $\mathbf{S}$ , the 3-D matrix of spends by all shoppers across product categories and across months. We unroll the dimensions along  $p$  (product categories) and  $m$  (months) into a single dimension of size  $|p| \times |m|$ .

We now consider each cluster separately. Let  $\mathbf{X}$  refer to the resulting sparse 2-D matrix of spends of shoppers within a cluster. Let  $\tau_1$  and  $\tau_0$  represent indices corresponding to known and unknown values respectively. The non-zero data values of the matrix,  $\mathbf{X}_{\tau_1}$ , have normal distributions that make SVD very suitable for modeling. Depending on the goal of the analysis, the unknown values  $\mathbf{X}_{\tau_0}$  can be viewed as data points with zeros or as missing values.

Treating these as missing values and imputing values using the SVD gives us an estimate that can be interpreted as how much a customer would buy if they chose to meet that shopping need. Again, this is analogous to the approach taken in movie recommender problems. We use IISVD for the imputation and compute full-rank decompositions. Given a sparse matrix  $\mathbf{X}$ , IISVD begins by reordering the rows and columns of an initial sample of data to form the largest dense matrix possible. SVD is performed on this dense data and is then iteratively “grown” step by step as more rows and columns are added. IISVD provides efficient and accurate algorithms for performing rank-1 modifications to an existing SVD such as updating, downdating, revising and recentering. It also provides efficient ways to compute approximate fixed-rank updates, see [2] for a more detailed treatment. This process is repeated for each of the 23 clusters of shoppers separately and imputed values  $\hat{\mathbf{X}}_{\tau_0}$  are obtained. The imputed spends for a shopper are equivalent to a linear mixture of the spends of all shoppers within the cluster, weighted by the correlations between their spends and spends of the current shopper.

Note that if we now use this filled data set and impute the value of a known data point, the imputed values of the missing data have no effect on the solution because they already lie on the SVD regression hyperplane.

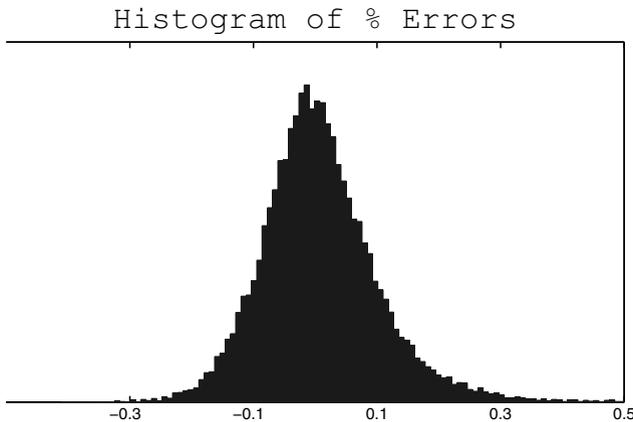
### Headroom Model

The next step is to measure of how each shopper is over-spending or under-spending in each  $L_3$  subcategory. Overspending corresponds to the degree by which a shop-

per’s spend differs from the expected value given by the SVD imputation of non-zero data.

Using the filled shopper- $L_3$  spend data  $\hat{\mathbf{X}}_{\tau_0}$ , we remove 10% of the known data values and impute new values using a thin SVD algorithm [2]. The thin SVD method we use determines the optimal rank for minimizing the error of imputation by way of a cross validation method. Because of this, the reduced rank used to model each cluster and even differing portions of the same cluster can vary. A result of this as shown in [4] is that the error of a model aggregated from these parts has much lower error than modeling all of the data at the same rank. This process is repeated for all known spend values  $\mathbf{X}_{\tau_1}$  and model estimates  $\hat{\mathbf{X}}_{\tau_1}$  are calculated.

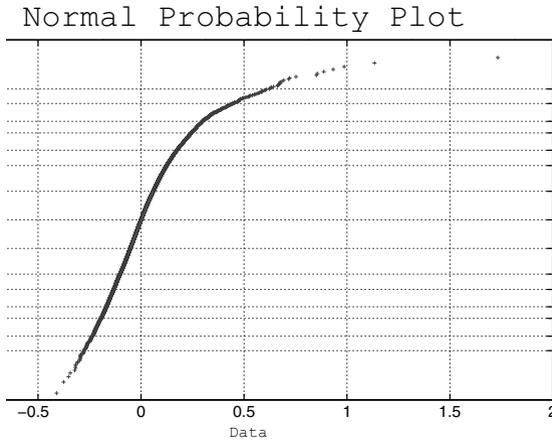
The residues, differences between known spends  $\mathbf{X}_{\tau_1}$  and modeled spends  $\hat{\mathbf{X}}_{\tau_1}$ , are normally distributed and hence can be expressed as Z-scores. Figure 1 illustrates the normality for percent errors in a given cluster. Figure 2 shows the normality plot of the data. The preponderance of points exhibit a normal distribution while the tail extremes diverge from normality.



**Fig. 1** Histogram of Percent Errors. The difference between the imputed spend values  $\hat{\mathbf{S}}_{\tau_1}$  and known spend values  $\mathbf{S}_{\tau_1}$  is expressed in terms of percentages. Figure shows that distribution is close to a Gaussian.

Observing the root-mean-squared error of the imputed known values  $\hat{\mathbf{X}}_{\tau_1}$  for each  $L_3$  category gives a clear quantification of the relative confidence we can have across  $L_3$  product categories.

The model  $z$ -scores generated in this process can be interpreted as a direct probabilistic measure of shopper over-spending/under-spending for each  $L_3$  category. One can do a similar analysis for SPI data and obtain  $z$ -scores in the same fashion. How-



**Fig. 2** A normality plot of the percent error data.

ever, it can be shown that the SPI  $z$ -scores obtained will be identical to the spend  $z$ -scores that we have calculated<sup>3</sup>.

### Consolidating Headroom Metrics

We chose to create a Consolidated Headroom Metric (CHM) from four individual Headroom Proxy Measures (HPM) for each shopper. The headroom model  $Z$ -scores described in the previous section is one of the four HPMs.

Two of the HPMs were computed directly from known spend and SPI data. For each shopper, known spend data  $S_{\tau_1}$  and known SPI data  $I_{\tau_1}$  were both expressed as  $z$ -scores for each  $L_3$  category. For both spend and SPI data, all the  $L_3$   $z$ -scores for each shopper were summed, weighted by the percentage spends of the shopper across  $L_3$  categories. The resulting  $z$ -scores are the Spend Headroom Metric and the SPI Headroom Metric respectively.

Lastly, we included the shopping frequency of each shopper. Although not a strong proxy for customer headroom, it is of value in identifying which customers with headroom to pursue for targeted marketing and promotions. By determining the probability of shopping frequencies from the frequency probability distribution function, we can map each shopping frequency to a  $z$ -score, which then acts as the fourth HPM. This enables us to combine information across all four of our headroom proxy measures.

<sup>3</sup> We model the spend in categories that a customer has shopped in by making a reasonable assumption that the number of items of different products bought will not change. Thus any increase/decrease in total spend is equivalent in percentage terms to the increase/decrease in SPI.

For each of the HPMS, we select the shoppers corresponding to the top 30% of the z-score values. The union of these sets is identified as the screened set of customers with the greatest likelihood of exhibiting headroom.

The Consolidated Headroom Metric for each shopper is created by a weighted sum across each of our four HPMS. In this project, we chose to apply equal weights and computed the CHM as the mean of the four HPMS. However, one could subjectively choose to emphasize different HPMS depending on the analysis goals.

## 6 Conclusions

In this paper, we presented the case-study of a retail data mining project. The goal of the project was to identify shoppers who exhibit headroom and target them with product categories they would most likely spend on. We described details of the SVD-based recommender system and showed how to identify customers with high potential spending ability. Due to the customers' demand for mass customization in recent years, it has become increasingly critical for retailers to be a step ahead by better understanding consumer needs and by being able to offer promotions/products that would interest them. The system proposed in this paper is a first step in that endeavor. Based on the results of a similar highly successful project that we completed for another retailer previously [4], we believe that with a few iterations of this process and fine tuning based on feedback from sales and promotions performance, it can be developed into a sophisticated and valuable retail tool.

**Acknowledgements** The authors would like to thank Gil Jeffer for his database expertise and valuable assistance in data exploration.

## References

1. M. Brand. Incremental Singular Value Decomposition of Uncertain Data with Missing Values. In *European Conf on Computer Vision*, 2002.
2. M. Brand. Fast Online SVD Revisions for Lightweight Recommender Systems. In *SIAM Intl Conf on Data Mining*, 2003.
3. S. Chandrasekaran, B. Manjunath, Y. Wang, J. Winkeler, and H. Zhang. An Eigenspace Update Algorithm for Image Analysis. *Graphical Models and Image Proc.*, 59(5):321–332, 1997.
4. M. Giering. Retail sales prediction and item recommendations using customer demographics at store level. In *KDD 2008 (submitted)*.
5. M. Gu and S. Eisenstat. A Stable and Fast Algorithm for Updating the Singular Value Decomposition. Technical Report YALEU/DCS/RR-966, Yale University, 1994.
6. D. Gupta and K. Goldberg. Jester 2.0: A linear time collaborative filtering algorithm applied to jokes. In *Proc. of SIGIR*, 1999.
7. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender system - a case study. In *Web Mining for ECommerce*, 2000.