SVD Filtered Temporal Usage Pattern Analysis and Clustering

Liang Xie *
Reliant Energy LLC., NRG

Aug 31, 2009

Abstract

This paper examines a SVD-based effective and robust approach to discover temporal electricity usage pattern from a broad customer base. The rationale behind this method and the relationship between SVD, PCA and Fourier analysis is discussed, as well as the implementation of SVD in SAS without IML modular. The output from this method is taken as input to the k-means clustering algorithm in PROC FASTCLUS. Using synthetic data, we demonstrate how robust the filtered data is to outliers and how effective the k-means algorithm can leverage the regulated data.

1 Introduction

Recent development in advanced meter provides residential electricity retailers unprecedent opportunity to collect and analyze high frequency usage data from each individual customers and offers enormous information that challenges analysts and IT infrastructure.

We are facing two fundatmental difficulties in processing and analyzing such data.

First of all, the amount of data is huge. During the past six months, from less than 50K currently visible smart meter customers, we cumulated over 40GB usage data for 15min interval and 60min interval alone, and this data grows at an increasing rate. Since April, 2009, the data increases at 15GB to 20GB per month. Given the fact that Reliant has over 1.7 million residential and small business customers and all of them will be equipped with advanced meter in the next couple of years under DoE's plan, we are facing a serious problem. This issue just becomes worse if ERCOT decides that all competitive electricity retailers have visibility to all Texas residential customers that equipped with advanced meters.

Second, due to the enormous amount of data, many fancy models for time series and panel data are litterally not feasible. We need a simple yet effective method to discover usage patterns that are of business interests

*Correspondence Author

and of operation feasibility. This method should be robust to outliers and be fast to execute.

In this paper we introduce a robust approach based on SVD to tackle this business problem. We will first discuss the rationale behind using SVD by exploring the relationship between SVD and Fourier analysis. Then we demonstrate how SVD filter can transform original synthetic unorganized data into regulated one that shows meaningful patterns. Finally, its effectiveness and robustness are discuss by applying a k-means algorithm to this filtered data.

2 SVD Definition

SVD, aka Singular Value Decomposition, is an important mathematical tool to factor rectangular matrix and has its wide applications in areas such Data Compression, Pattern Recognization, Microarray analysis, Signal Processing, etc. It is based on the following theorem in linear algebra [2]: Any m-by-n matrix X can be written as the product of an m-by-n unitary matrix U, an n-by-n diagonal matrix Σ with positive or zero elements (the singular values), and the transpose of an n-by-n unitary matrix V. In mathematical form:

$$X = U\Sigma V^T \tag{1}$$

Both U and V are unitary matrices in the sense that that $UU^T = I,\ VV^T = I$, which also means both matrices are column-orthogonal. Because matrix V is square, V is also row-orthogonal. Expand formula (1), we have

$$X_{ij} = \sum_k u_{ik}\sigma_k v^T_{kj} = \sum_k c_{jk} v^T_{kj} \tag{2}$$

This formula resembles the Fourier analysis in the sense that the cyclical term: $\exp \iota 2\pi jk/m$ is replaced by the normalized eigen vector term v^T_{kj}. However, while the U matrix from SVD are orthogonal, the coefficient matrix $C_{[ij]}$ from Fourier analysis is not necessarily orthogonal. That is SVD can be thought as a special Fourier analysis where the cyclical basis terms are determined in a particular way according to the linear algebra definition.

Matrix U can be regarded as an "expression" basis vector directions for X and V is the corresponding "profile" basis vector directions. Therefore, each row and column of these matrices provide special information regarding the original matrix X.

Any row $\mathbf{x}_{i\cdot}$ of the matrix X can be expressed as:

$$\mathbf{x}_{i\cdot} = \sum_{k=1}^{n} u_{ik}\sigma_k \mathbf{v}^T_{k\cdot} \tag{3}$$

Any column $\mathbf{x}_{\cdot j}$ of the matrix X can be expressed as:

$$\mathbf{x}_{\cdot j} = \sum_{k=1}^{n} \mathbf{u}_{\cdot k}\sigma_k \mathbf{v}^T_{kj} \tag{4}$$

Therefore, each row of X is a linear combination of basis profile data and each column of X is a linear combination of basis expression data.

3 SVD in SAS®

SAS® doesn't explicitly support SVD except for in its IML modular which many companies do not license. However, we are able to trick SAS® in order to get a solution that is the same as the SVD output upto a scalar constant. This trick relies on one of many ways SVD is calculated and its direct relationship with PCA. Actually, SVD and PCA are so closely related that in literature, they are also alway discussed inter-changablly.

Because output matrix U is the same dimension as original matrix X and both U and V are unitary, if original matrix X is square, we know the matrix U will be the same as matrix V:

$$X^T X = VSV^T \tag{5}$$

where $S = \Sigma^2$. Once we get both S and V, the output matrix U can be calculated as follows:

$$U = XVS^{-1} \tag{6}$$

We found these steps are very similar to how PCA was calculated where the covariance matrix of X is used. Then, if we conduct a PCA on the original matrix X based on uncorrected covariance matrix, the right eigen-matrix is the V matrix, and the diagonal elements in the eigen value matrix S is the square of the corresponding elements in the singular value matrix Σ, up to a scalar constant, which is the square root of the row dimension of X. This can be seen by inspecting formula (5). The uncorrected covarance matrix used in SAS PROC PRINCOMP is actually $X^T X/n$, where n is the number of observations, i.e. row dimension of X. Therefore S in formula (5) is actually n times of the eigen values from PROC PRINCOMP output. With all these matrices at hand, we can employ formula (6) to get matrix U.

Therefore, we can employ the PROC PRINCOMP procedure in SAS to accomplish SVD. As an example, suppose we have a matrix:

$$\mathbf{X} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \\ 17 & 18 & 19 & 20 \end{pmatrix}$$

, its SVD output obtained from R is:

$$\mathbf{U} = \begin{pmatrix} -0.09655 & -0.76856 & -0.63199 & 0.005024 \\ -0.24552 & -0.48961 & 0.63930 & 0.536804 \\ -0.39448 & -0.21067 & 0.29155 & -0.665307 \\ -0.54345 & 0.06827 & 0.02697 & -0.299895 \\ -0.69242 & 0.34721 & -0.32582 & 0.423374 \end{pmatrix}$$

$$\mathbf{V} = \begin{pmatrix} -0.4430 & 0.7097 & -0.03457 & -0.5466 \\ -0.4799 & 0.2640 & 0.45353 & 0.7031 \\ -0.5167 & -0.1816 & -0.80335 & 0.2337 \\ -0.5536 & -0.6273 & 0.38439 & -0.3902 \end{pmatrix}$$

$$\mathbf{\Sigma} = \mathrm{diag}\,\{5.352\text{E+01},\;\; 2.363\text{E+00},\;\; 3.654\text{E-15},\;\; 5.646\text{E-16}\}$$

Submit the following code to SAS:

```
proc princomp data=X
              outstat=Xout(where=(_type_ in ('EIGENVAL', 'USCORE')))
              noint cov noprint;
     var X1-X3;
run;
```

where the row named "EIGENVAL" contains elements of S, and the rows of "USCORE" contains the transposed eigen-vector matrix V^T. You can obtain these quantities via ODS, too. For example:

```
ods output EigenValues=S;
ods output EigenVectors=V;
```

Note that, first, unlike the OUTSTAT statement, the eigen-vector matrix obtained from ODS OUTPUT is not transposed; second, the square roots of calculated S elements are upto the scalar constant of square root of number of observations, comparing to SVD output. The SAS output of right eigen-vector matrix V and eigenvalue matrix S are, respectively:

$$\mathbf{V} = \begin{pmatrix} 0.4430 & 0.7097 & -0.1195 & -0.5345 \\ 0.4799 & 0.2640 & -0.2390 & 0.8018 \\ 0.5167 & -0.1816 & 0.8367 & 0.0000 \\ 0.5536 & -0.6273 & -0.4781 & -0.2673 \end{pmatrix}$$

$$\mathbf{S} = \mathrm{diag}\,\{572.883,\; 1.117,\;\; 0.000,\;\; 0.000\}$$

We see the first two axles of V flip comparing to the R output, but it doesn't matter since SVD is only unique upto performing an orthogonal rotation on any set of columns of U and V whose corresponding elements of S happen to be exactly equal [2]. Besides, the columns of eigen-vector matrix are numericall correct upto the *significant* eigenvalues. This, while not so desirable, is not of practical importance because only those eigen vectors of *significant* eigenvalues matter.

If we scale the eigenvalues by dividing by the number of observations and taking square root, we can obtain the singular values from SVD:

$$\mathbf{\Sigma} = \mathrm{diag}\,\{53.520\;\; 2.363,\;\; 0.000,\;\; 0.000\;\}$$

In order to get U from SAS®, we apply formula (6) in SAS®DATA STEP and get the result that is the same as SVD output corresponding to *significant* singular values:

$$\mathbf{U}^{(\mathbf{SAS})} = \begin{pmatrix} 0.09665 & -0.76856 & 0 & 0 \\ 0.24552 & -0.48961 & 0 & 0 \\ 0.39448 & -0.21067 & 0 & 0 \\ 0.54345 & 0.06827 & 0 & 0 \\ 0.69242 & 0.34721 & 0 & 0 \end{pmatrix}$$

SVD is a robust mathematical tool and SAS is highly capable to conduct SVD computation with large data set. According to SAS Online

Documentation, the time required for full computation is roughly at the order of $\mathcal{O}(V^3)$ and the memory required is at the order of $\mathcal{O}(V^2)$, where V is the number of variables included for computation. On a PC equipped with C2D6320 1.86GHz CPU and 3GB RAM, we test the algorithm on a data set with 1.7 million observations and 400 variables, and the SAS log shows:

```
15   data x;
16           length ID 7;
17           array _X{*}  X1-X400;
18           do ID=1 to 1.7E6;
19               do j=1 to 400;
20                  _X[j]=ranuni(1000);
21               end;
22               output;
23           end;
24   run;

NOTE: The data set WORK.X has 1700000 observations and 402 variables.
NOTE: DATA statement used (Total process time):
      real time           2:57.60
      cpu time            1:34.24


25
26
27   proc princomp data=x  noint cov noprint  outstat=_stat_;
28           var X1-X400;
29   run;

NOTE: The NOINT option causes all moments to be computed about the origin.
      Correlations, covariances, standard deviations, etc., are not corrected
      for the mean.
NOTE: The data set WORK._STAT_ has 803 observations and 402 variables.
NOTE: PROCEDURE PRINCOMP used (Total process time):
      real time           7:56.63
      cpu time            5:52.00
```

Now we found a way to get the exact SVD output from from tweaking PCA, which opens a door to many data mining algorithms that are SVD-based, such as gene-expression analysis [3], Latent Semantic Index in Text Mining area [6], etc.

4 Temporal Usage Pattern Analysis

Temporal electricity usage data are highly skewed and very volatile in nature, which cause serious problems when using clustering algorithms on these naive data. In this section, we show how we can utilize the SVD filter to accommondate outliers, smooth data and extract business valuable information from massive household level electricity usage data.

With implementation of Advanced Meters, also called Smart Meter, in Texas, eletricity retail houses begin to collect huge amount of electricity usage data at 15 minutes, 60 minutes and daily intervals for each household equipped with advanced meter.

After tedious data cleaning, which deseres a separate paper, we come to 60-min interval records in continuous 6 month of about 20K smart meters, that is about 86million observations. Due to limitation in computating power (currently we are using a poor PC for data processing and analysis), we randomly selected 5K meters which has continuous 6 months service for further analysis. The data has two layers' information. The first layer is daily eletricity usage over the 6 continuously observed months, the other is hourly eletricity usage within a day across the 6 months. We developed an Multilayer SVD-filtered K-means clustering algorithm to analyze this data. This algorithm is based on the one detailed by Alter et al. [3]. Our algorithm follows these steps:

1. For each meter, obtain their daily eletricity usage as a percentage across the 6 month total; Pool all meters together to form a panel data; This step normalizes the usage of each meter to 1 which equivalently controls the so called steady state variance;
2. In SAS, apply SVD to above panel data via PCA on non-centered covariance matrix, obtaining matrices $V, \sqrt{S}$
3. Using formula 6 to get matrix U
4. Set the first eigen value proportion in $\sqrt{S}$ to be 0, and use definition formula 1 to get matrix X'; In observing the relationship between SVD and Fourier Analysis, this matrix can be seen as a low pass filtered result. Alter et al. claimed that this step removes the steadstate mean. Note that in Alter et al.'s paper, they conducted two SVD filtering process, respectively, to remove the steady state mean and steady state variance; But our normalization step 1 can achieve similar effect;
5. Apply SVD again to low pass filtered matrix X', calculate the correlation of each meter's profile to eigen profile vectors in matrix V; This correlation serves as a similarity index for each day and as features for K-means clustering algorithm;
6. Apply k-means clustering algorithm (PROC FASTCLUS) using these correlations to calculate the distance measures. There are several methods proposed to select the optimal number of clusters, K [1]. We employ Bayesian Information Criteria (BIC) for its simplicity and robustness. Optimal K corresponds to the min BIC value; The BIC formula follows Moor [5]:

   $$BIC = Distortion + k * (\text{num of variables}) * \log(N) \tag{7}$$

   where $Distortion = \sum_k \sum_{i \in \Omega(\bar{\mathbf{c}}_\mathbf{k})} (\mathbf{x_i} - \bar{\mathbf{c}}_\mathbf{k})^2$, $\bar{\mathbf{c}}_\mathbf{k}$ is the mean vector of cluster k, and $\Omega(\bar{\mathbf{c}}_\mathbf{k})$ is the set of points centred at $\bar{\mathbf{c}}_\mathbf{k}$. Distortion is obviously the sum of total variance of each cluster and can be readily obtained from PROC FASTCLUS in SAS®.

   But BIC critria applies to a given number of features so that we need to find a way to determine appropriate number of features to use. Unlike using naive raw data where the number of feature equals the number of variables or number of selected business meaningful variables, it is not clear how many to choose from the correlation features. In this project, we employed a MODE approach. Specifically, we choose only a small number of correlations such that the corresponding eigen values accounts for, say 70% to 85%, sum of all eigen values. This range is choosen aribitrary based on business understanding of underlying noise level. Then there is a corresponding range of number of features and for each number, we ran the BIC selection process, then we get the MODE of BIC selected optimal number of clusters in this iteration and set this K as the optimal one.

7. Within each identified daily level cluster, we obtain their hourly eletricity usage percentage across a "typical" day, or "typical" weekday and weekend. By "typical", we actually mean that the hourly percentage is the average over a month. So that we have 6 vectors of length 24, or 48 if separating weekday and weekend, for each meter and by pooling them together, we have a panel data of 144 or 288 columns for each meter during 6 months;
8. Within each identified daily level cluster, we repeat step 1 to 5 on each of the 6 monthly subsets of the panel data to get 6 months' correlation vectors for each meter;
9. We then choose first several correlation coefficients from each vector and conduct the k-means clustering analysis;
10. The appropriate number of clusters is determined via the same BIC approach;

This multilayer approach leverage the fact that daily usage pattern and hourly usage pattern within a day are driven by different factors. The daily usage pattern across months is mostly determined by seasonality and weather related factors, such as eletricity heating or gas heating, and house insulation, etc. On the other hand, hourly usage pattern within a typical day is mostly driven by the life style of the household. For example, singles will have very different usage curve in a day comparing to an established household where wife stays at home. Besides, since this algorithm applies low pass filtering up front, directly work on hourly usage pattern across all months won't work because the high frequent hourly amplitude is removed. We show an example using synthetic data that resemble typical scenarios in our study. The data is compiled in such a way that it is able to emphasize the feature of this algorithm.

4.1 Data Simulation

We use a synthetic data set to demonstrate this algorithm. Synthetic data is generated via a Mixed Model approach like in Wang and Tobias [7], where we assumed threed underlying clusters. In reality, it is very unlikely, if not impossible, that there are three distinctive groups but a continuum of patterns. So that a probablistic than a deterministic approach is more appropriate. To mimic this continuous nature of patterns, we introduced a transition smoothing parameter in the model, so that the final model is similar to the heterogenous model of Verbeke and Molenberghs [8] but the weights of mixture probability change graduately from one group to the other. The Model can be written as:

$$\log(Y_{it}) \sim \sum_{j=1}^{3} \alpha_{ik} N(\mathbf{X}_{kt} + Z_{it}\mathbf{u}_{kt}, V_k) \tag{8}$$

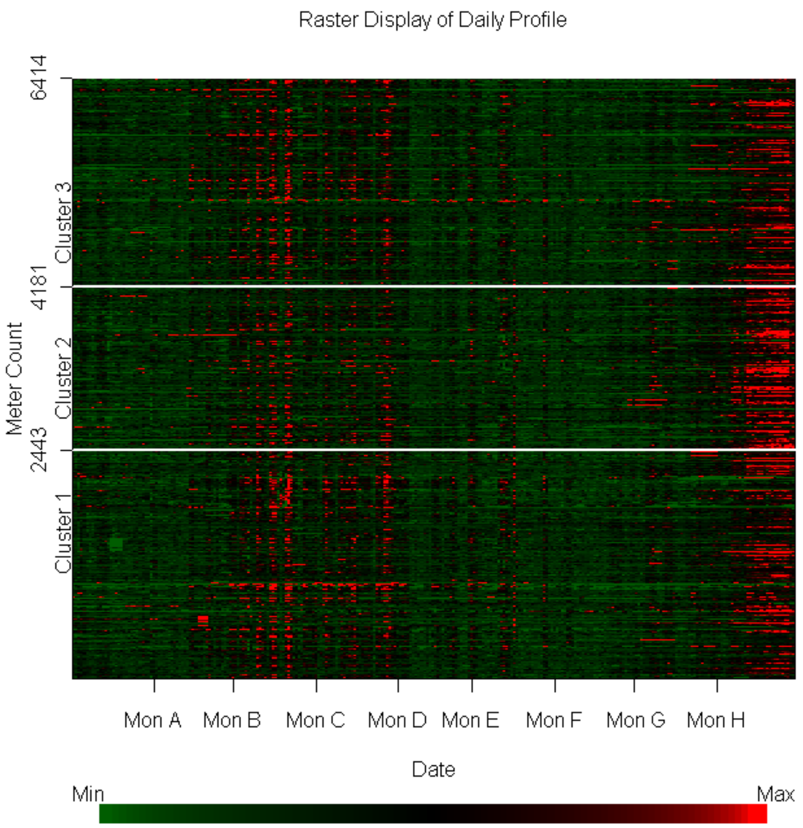

Figure 4.1: *Raster Display of Unclustered Daily Profile*

where α_{ik} is the weights parameter that changes with each simulated meter so that to smooth the transition from one cluster to the other. Note that in the original Heterogeneous Linear Mixed Model, $\alpha_{ik} = \alpha_k, \forall i$.

For details of the model, please consult [8]. Due to sensitivity of business information, we do not show the actual parameters and data we used in the model but related figures, and the data shown in this paper has been distorted by scaling and shifting. Figure 4.1 shows a Raster Display of the simulated raw data, not clustered. We observe two prominent usage peak across the time frame, in the middle and end, respectively.

4.2 Experiment Result

For demonstration purpose, we show here the clustering result following step 1 to 6 discussed above. Our data is a continuum of 3 latent classes, and using the MODE selection approach, we obtain the correct number of clusters. The clustering result has several desirable properties. As shown in Figure 4.2, where the cluster and data within a cluster is sorted by the correlation with the first eigenvector of second SVD filtering process, and we observed a noticable transition from Peak Usage during Month B, C, D to Month G, H. So that the eigenvectors are business meaningful and are able to provide analysts guidance on business intelligence interpretation. Secondly, from Figure 4.3, we found the clusters are mostly separable on their peak profile time window. The Boxplot shows that cluster 1 is well separated from the other 2 clusters from Month B to Month D, since the notch of cluster 1 is almost non-overlap to the notches of the rest clusters, and cluster 3 is well separated from cluster 1 and 2 from Month G and H. The non-overlapping of boxplot's notches is a strong indication of medians differ among the clusters, see [9]. Thirdly, even though the raw data has several outliers in the sense that in some time windows, their profile shows extreme values, the SVD filtered clustering result is pretty robust and those outlier observations won't occupy single clusters which would, however, be the case if we use raw data for k-means clustering. Under our algorithm, each cluster has adequant number of relatively homogeneous observations, which make this process well suited for automation.

5 Conclusion

In this paper, a SVD-based filter is applied to massive dense residential eletricity usage data over time. The filtered data is transformed into correlation between the filtered value and corresponding eigenvectors, then a k-means algorithm is applied to this correlation data using correlation with eigenvectors as features. Through SVD filtering and correlation analysis, we uncovered hidden patterns that not immediately available to business managers and disjoint clustering algorithm on correlation data is able to group customers into business meaningful segments while immune to outliers. The algorithm is designed towards facilitating automation. The output provides marketing team the capability to develop individualized heterogeneous pricing plans for each customer.

SAS program is avaiable upon request.

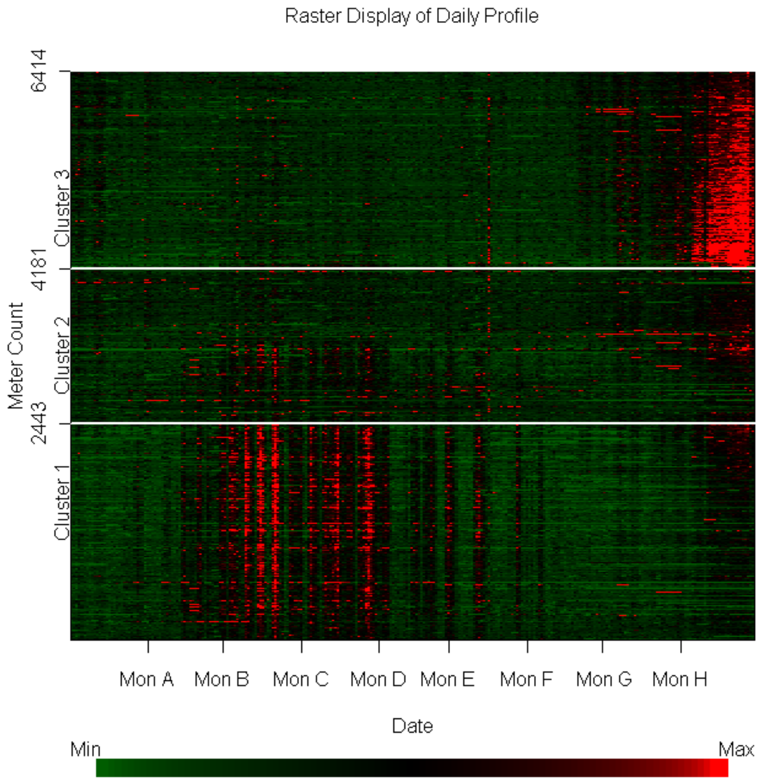

Figure 4.2: *Sorted Raster Display of Clustered Daily Profile*

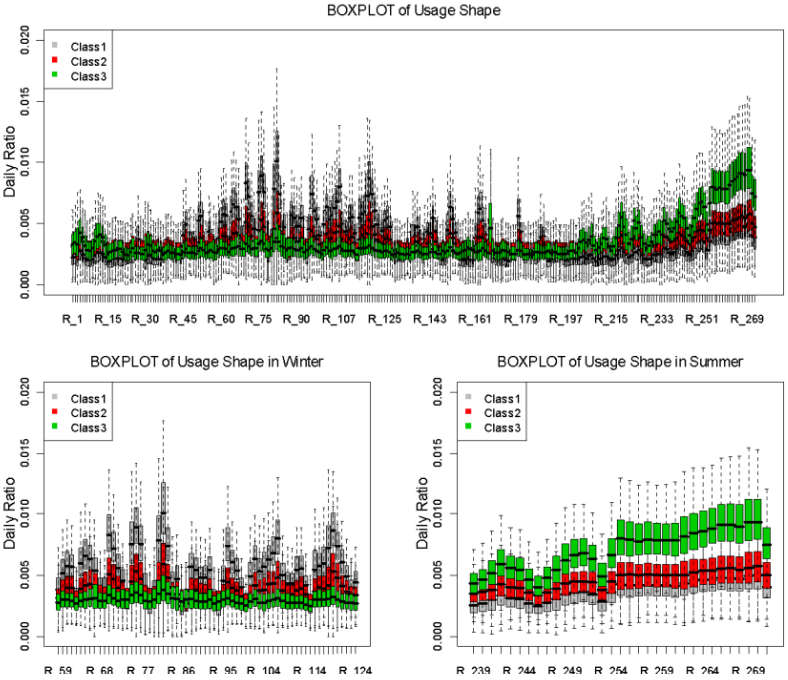

Figure 4.3: *Boxplot of Clusters' Profiles*

6 Reference

1. Friedman, Jerome, Hastie, Trevor, Tibshirani, Robert; *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer Series in Statistics, 2001
2. William Press, Saul Teukolsky, William Vetterling, Brain Flannery; *Numerical Recipts: The Art of Scientific Computing*, 3rd Edition, Cambridge University Press, 2007
3. Orly Alter, Patrick Brown, David Botstein; *Singular Value Decomposition of Genome-Wide Expression Data Analysis*, Proceedings of the National Academy of Science, vol.97, no.18, 2000
4. Michael Wall, Andreas Rechtsteiner, Luis Rocha; *Singular Value Decomposition and Principle Component Analysis*, In *A Practical Approach to Microarray Data Analysis*, Springer, 2003
5. Moore, Andrew; *K-means and Hierarchical Clustering*, Tutorial Slides, `http://www.autonlab.org/tutorials/kmeans.html`
6. Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R. A.; *Indexing by Latent Semantic Analysis*, Journal of the Society for Information Science, vol.41, no.6, 1990
7. Tianlin Wang, Randy Tobias; *All the Cows in Canada: Massive Mixed Modeling with the HPMIXED Procedure in SAS 9.2*, SGF2009, Paper Paper 256-2009, SAS Institute Inc.
8. Geert Verbeke and Geert Molenberghs; *Linear Mixed Models for Longitudinal Data*, 2nd Edition, Springer Series in Statistics, 2000
9. Chambers, J. M., Cleveland, W. S., Kleiner, B. and Tukey, P. A. *Graphical Methods for Data Analysis*; Wadsworth and Brooks / Cole, 1983

7 Contact Information

Liang Xie
Reliant Energy
1000 Main St
Houston, TX 77081

Work phone: 713-497-6908
E-mail: xie1978@yahoo.com
Web: www.linkedin.com/in/liangxie
Blog: sas-programming.blogspot.com