

Feature Extraction Methods for Time Series Data in SAS® Enterprise Miner™

Taiyeong Lee, Ruiwen Zhang, Yongqiao Xiao, and Jared Dean
SAS Institute Inc.

ABSTRACT

Because time series data have a unique data structure, it is not easy to apply some existing data mining tools directly to the data. For example, in classification and clustering problems, each time point is often considered a variable and each time series is considered an observation. As the time dimension increases, the number of variables also increases, in proportion to the time dimension. Therefore, data mining tasks require some feature extraction techniques to summarize each time series in a form that has a significantly lower dimension. This paper describes various feature extraction methods for time series data that are implemented in SAS® Enterprise Miner™.

INTRODUCTION

Time series data mining has four major tasks: clustering, indexing, classification, and segmentation. Clustering finds groups of time series that have similar patterns. Indexing finds similar time series in order, given a query series. Classification assigns each time series to a known category by using a trained model. Segmentation partitions time series. Time series data can be considered multidimensional data; this means that there is one observation per time unit, and each time unit makes one dimension. Some common data mining techniques are used for the tasks. However, in the real world, each time series is usually a high-dimensional sequence, such as stock market data whose prices change over time or data collected by a medical device. When the time unit is seconds, the dimensions of data that accumulate in just one hour are 3,600. Moreover, the dimensions are highly correlated to one another. The number of variables in training data increases proportionally as the time dimension increases. Most existing data mining tools cannot be used efficiently on time series data without a dimension reduction. Therefore, a dimension reduction is required through feature extraction techniques that map each time series to a lower-dimensional space. The most common techniques of dimension reduction in time series are singular value decomposition (SVD), discrete Fourier transformation (DFT), discrete wavelet transformation (DWT), and line segment methods. Some classical time series analyses, such as seasonal, trend, seasonal decomposition, and correlation analyses, are also used for feature extraction.

This paper shows how to use SAS Enterprise Miner to implement the feature extraction techniques. But it does not attempt to explain these time series techniques in detail and instead invites the interested reader to explore some of the literature on the subject: Keogh and Pazzani (2000a), Keogh et al. (2000), and Gavrilov et al. (2000) for time series dimensionality reduction and the TIMESERIES procedure documentation in SAS/ETS® for the classical time series analyses.

This paper is organized as follows. First, it explains feature extraction and dimension reduction techniques that use classical time series analysis, and then it describes feature extraction techniques that use some well-known mathematical functions for dimension reduction. Next, it demonstrates the performance of dimension reduction techniques by presenting some examples. The paper uses SAS Enterprise Miner 13.1 to demonstrate these techniques.

FEATURE EXTRACTION USING CLASSICAL TIME SERIES ANALYSIS

TIME SERIES REDUCTION WITH TIME INTERVALS

Summarizing the time series based on a specific time interval is one of the simplest data reduction methods. Figure 1 shows data reduction plots of the airline passenger data (monthly, from January 1949 to December 1960) in the Sashelp library based on several time intervals. The data set contains 144 monthly data points (12 months \times 12 years). While the plots move from high frequency to low frequency in the time interval, the data points are reduced and the trends are shown.

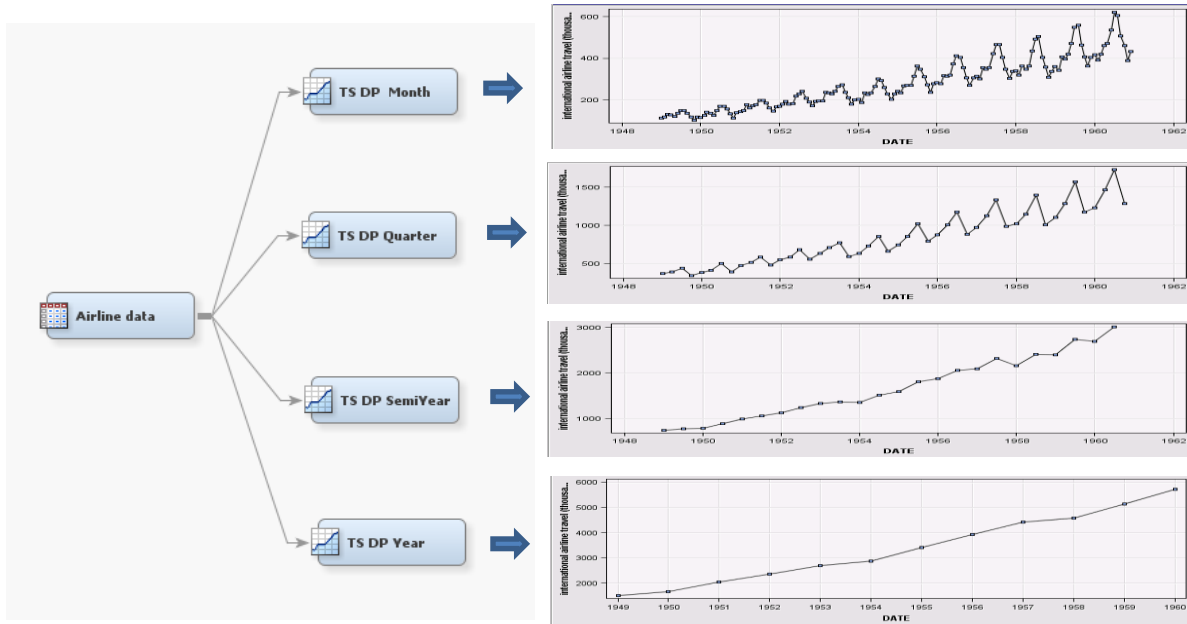


Figure 1. Reducing Time Series Based on Time Frequencies

Trend analysis is a line segment dimension reduction method that is explained later, but this type of line segment occurs based on time frequencies only. Arbitrary line segmentation is not allowed. You can use the trend analysis as a pre-data cleaning procedure that includes accumulation, transformation, differencing, and missing value replacement.

SEASONAL ANALYSIS

Seasonal analysis provides five statistics: sum, mean, median, minimum, and maximum. For a seasonal cycle length that is implicitly given by the time interval, the statistics are calculated at each season point over the seasonal cycle. Suppose $\{x_1, x_2, \dots, x_{120}\}$, and the seasonal cycle is 4. Using mean statistics, you obtain a set of $\{s_1, s_2, s_3, s_4\}$ by seasonal analysis:

$$s_1 = \text{mean} \{x_1, x_5, x_9, \dots, x_{113}, x_{117}\}$$

$$s_2 = \text{mean} \{x_2, x_6, x_{10}, \dots, x_{114}, x_{118}\}$$

$$s_3 = \text{mean} \{x_3, x_7, x_{11}, \dots, x_{115}, x_{119}\}$$

$$s_4 = \text{mean} \{x_4, x_8, x_{12}, \dots, x_{116}, x_{120}\}$$

Therefore, the 120–time dimension space is mapped to the 4–dimension space. No matter how many time dimensions you have, the number of reduced dimensions is the seasonal cycle length.

Figures 2 and 3 show seasonal analyses of the cosmetic sales data in the SAS Sampsi library. The data set consists of three cross ID variables (**Group**, **State**, and **SKU**), a time ID variable (**Month_Yr**), and a target (**Sales**). It contains monthly sales data collected from January 1996 to December 1998. The **State** and **Group** variables are ignored in this example, so the sales data have been accumulated over the **State** and **Group** variables.

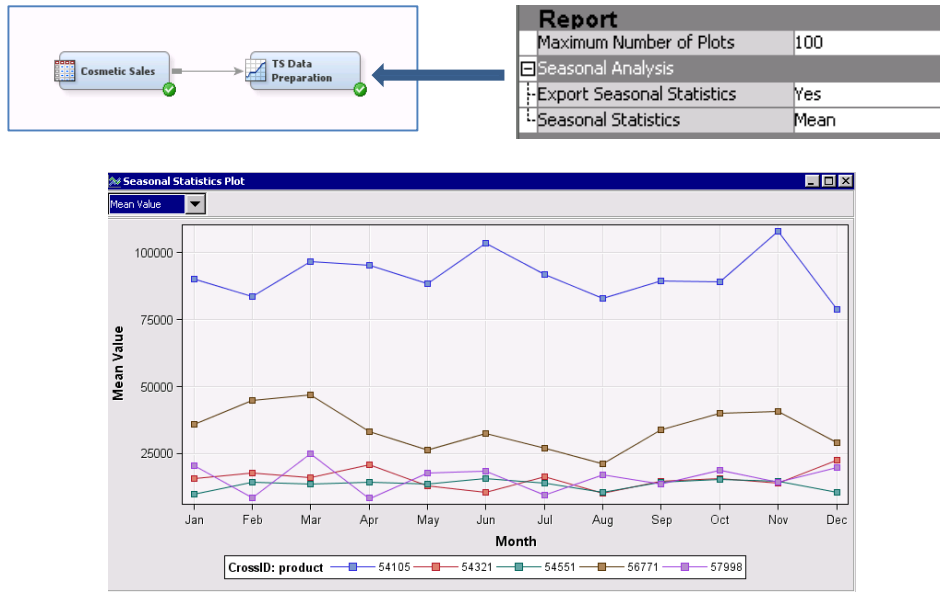


Figure 2. Seasonal Analysis of Cosmetic Sales Data

If you change the Time Interval property to Quarter, you get the output summarized by quarter.

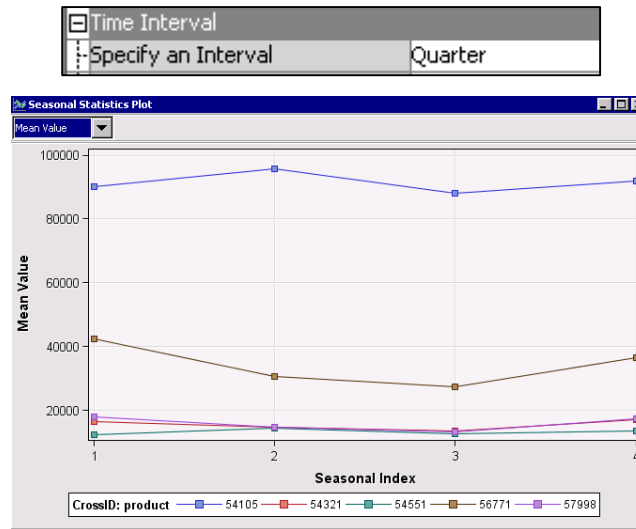


Figure 3. Seasonal Analysis of Cosmetic Sales Data

Overall, product 54105 has higher sales than the other products evenly over all the seasons. You can also use the season data for classification or clustering when you export the season statistics.

CORRELATION ANALYSIS

Correlation analysis provides autocorrelation and cross-correlation when target series are defined. The time series data can be characterized using time domain statistics such as autocovariances, autocorrelations, normalized autocorrelations, partial autocorrelations, normalized partial autocorrelations, inverse autocorrelations, normalized inverse autocorrelations, and white noise test statistics. These statistics are used for model identification in the ARIMA model, so they represent the feature of time series well. For example, AR processes can be identified by the shape of the autocorrelation function (ACF) plot and their order by the shape of the partial autocorrelation function (PACF) plot, and MA processes can be identified by the shape of the PACF plot and their order by the shape of the ACF plot. If you use the ACF and PACF together, the resulting features represent an ARIMA model well. Figure 4 shows how to use the correlation analysis for feature extraction. Both time series nodes use the maximum lag of 5, and you merge two outcome data sets into one data set for clustering input data. Using the Metadata node, you delete the lag 0 because its value is constant. The Cluster node has 10 input variables (5 from ACF and 5 from PACF) and the time series ID variable. You set the maximum number of clusters as 5.

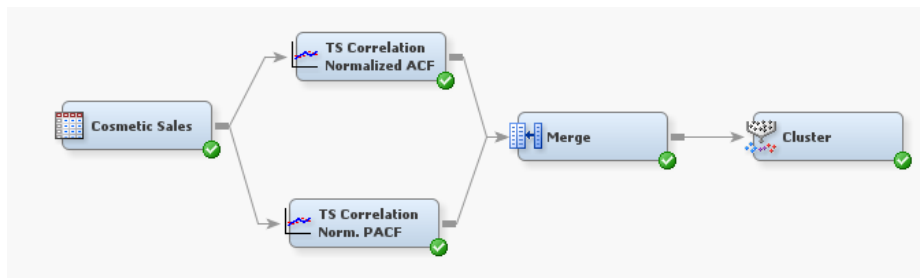


Figure 4. Clustering Flow Using Both ACF and PACF

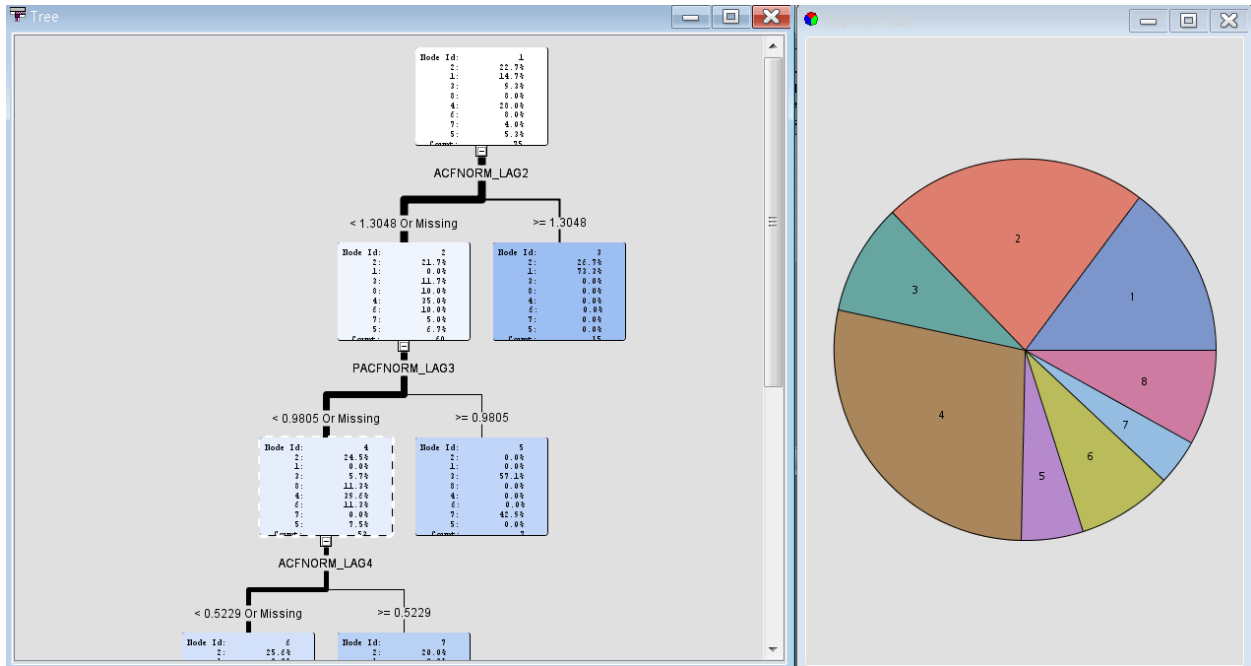


Figure 5. Tree Plot Resulting from the Cluster Node

Figure 5 shows apparently two variables: normalized ACFs at LAG2 and LAG4 and normalized PACF at LAG3 as key classification variables. Using both ARIMA model identification statistics could give better results in time series clustering. However, in order to have more precise clustering, it is necessary to do further analyses by using the fitted coefficients within the classified model category.

Note that the TS Correlation node also provides the functionality of time covariate selection based on the cross-correlation between target and input series.

SEASONAL DECOMPOSITION

You could use the classical seasonal decomposition techniques to divide each time series into several components, such as the trend-cycle component, seasonal irregular component, seasonal component, trend-cycle-seasonal component, irregular component, seasonally adjusted component, percent change seasonally adjusted component, trend component, and cycle component. The four commonly used seasonal decomposition techniques are as follows:

- Additive $O_t = (TC)_t + S_t + I_t$
- Multiplicative $O_t = (TC)_t S_t I_t$
- Log-additive $\log(O_t) = (TC)_t + S_t + I_t$
- Pseudo-additive $O_t = (TC)_t (S_t I_t - 1)$

In these equations, (O_t) is the original time series, (S_t) is the seasonal component, $(TC)_t$ is the trend-cycle component, and (I_t) is irregular components. You can further decompose the trend-cycle component $(TC)_t$ into separate trend (T_t) and cycle (C_t) components by using fixed-parameter filtering techniques. The multiplicative mode is for series that are strictly positive, and the pseudo-additive mode is for series that are nonnegative. For more

information about the seasonal decomposition formulas, see the TIMESERIES procedure documentation in SAS/ETS. Figure 6 shows a decomposition example that uses the airline passenger data set. It also shows the various types of decomposed series from the airline data.

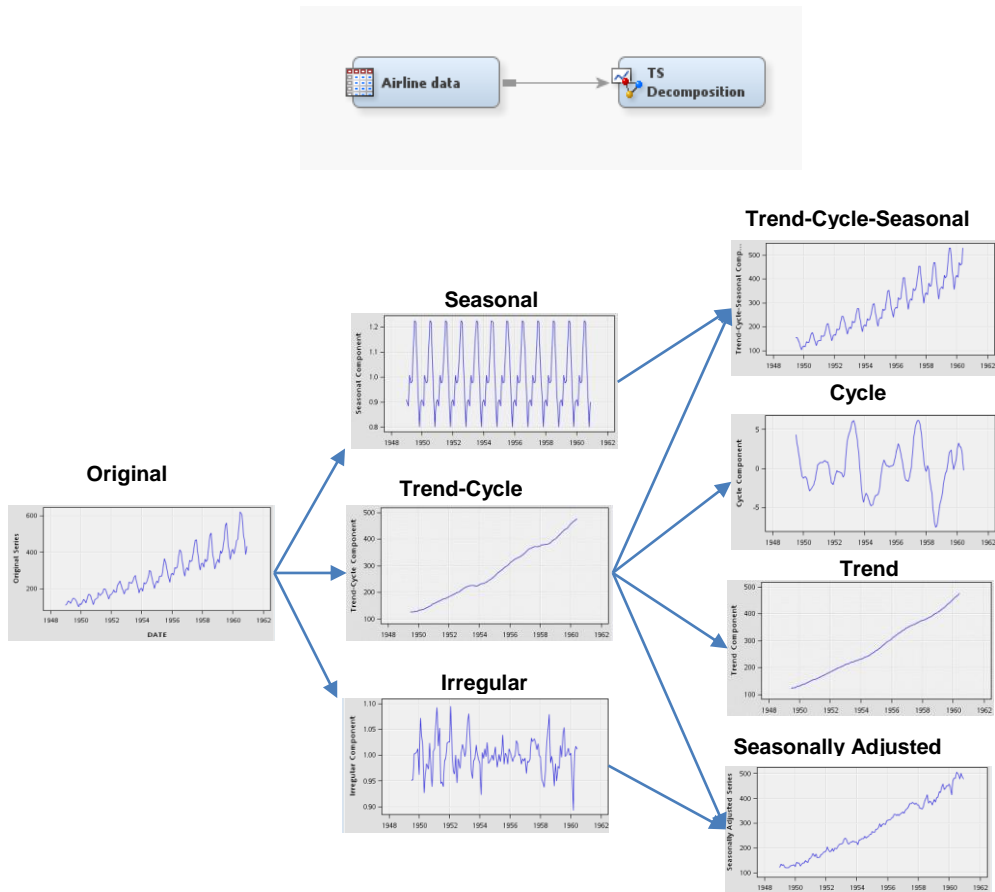
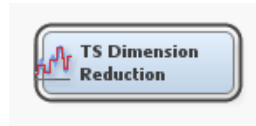


Figure 6. Decomposed Series from the Original Series

You can use the decomposed features of time series as inputs for the TS Dimension Reduction node and in further clustering analysis. An example is shown on page 11.

FEATURE EXTRACTION FOR DIMENSION REDUCTION

The TS Dimension Reduction node implements several feature extraction methods for time series dimension reduction that are described in the following sections. Figure 7 shows the property sheet of the TS Dimension Reduction node.



Variables	
[-] Normalization	
Apply Normalization	Yes
Normalization Method	Global Normalization
Piecewise Normalization Bins	10
[-] Dimension Reduction Method	
Reduction Method	Discrete Wavelet Transform
Dimension Number	10
Output Data Type	Clustering Input Data
Max Number of Time Series to Plot	100

Figure 7. Property Sheet of TS Dimension Reduction Node

Suppose that \mathbf{X} is an $m \times T$ matrix of m time series with length T . One of this paper's goals is to demonstrate how to reduce the matrix \mathbf{X} to a matrix \mathbf{Y} of dimensions $m \times d$, where $d < T$. When the time dimension is reduced, the classification tools of SAS Enterprise Miner can be used to cluster or classify the m rows of \mathbf{Y} .

SINGULAR VALUE DECOMPOSITION

Singular value decomposition (SVD) is a classical dimension reduction method that is used in a wide variety of statistical analyses. Suppose \mathbf{X} is an $m \times T$ matrix of m time series of length T . \mathbf{X} can be expressed as a singular value decomposition, $\mathbf{U} \times \mathbf{Q} \times \mathbf{V}$, where \mathbf{Q} is a diagonal matrix of the m eigenvalues of $\mathbf{X}'\mathbf{X}$ (or singular values of \mathbf{X}), \mathbf{U} is the $m \times T$ matrix of orthonormal eigenvectors vectors of $\mathbf{X}\mathbf{X}'$, and \mathbf{V} is the $T \times n$ matrix of orthonormal eigenvectors vectors of $\mathbf{X}'\mathbf{X}$. This section demonstrates how SVD can be used for time series data reduction.

Figure 8 shows the relationship between the original set of time series, the decomposition, and the reduction. To obtain the desired dimension reduction ($d < T$), choose only the d largest singular values and the associated eigenvectors of $\mathbf{X}\mathbf{X}'$ and $\mathbf{X}'\mathbf{X}$. Based on this selection, subset the \mathbf{Q} matrix to obtain a $d \times d$ diagonal matrix, \mathbf{D} , of the largest eigenvalues; subset the \mathbf{U} matrix to obtain an $n \times d$ matrix, \mathbf{U} , of eigenvectors; and subset the \mathbf{V} matrix to obtain a $d \times m$ matrix, \mathbf{V} , of eigenvectors.

The d row vectors become a basis for the dimension reduction technique. The $\mathbf{Y} = \mathbf{U} \times \mathbf{Q}$ matrix with dimensions $m \times d$ is the set of the reduced time series. If you use all the T eigenvalues ($d = T$), the reconstruction forms exactly the same \mathbf{X} matrix (no reduction).

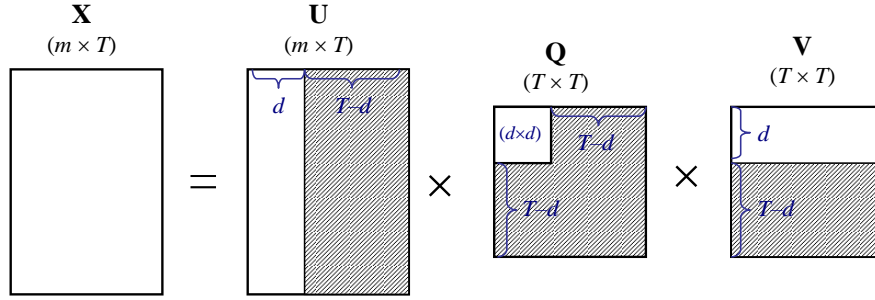


Figure 8. Singular Value Decomposition of X Matrix

In the TS Dimension Reduction node, SVD works as follows: it decomposes the $(m \times n)$ matrix \mathbf{X} (where m is greater than or equal to n) into the form $\mathbf{U} \text{diag}(\mathbf{Q}) \mathbf{V}^t$, where $\mathbf{U}^t \mathbf{U} = \mathbf{V}^t \mathbf{V} = \mathbf{V} \mathbf{V}^t = \mathbf{I}_n$, \mathbf{Q} contains the singular values of \mathbf{X} , \mathbf{U} is $m \times n$, \mathbf{Q} is $n \times 1$, and \mathbf{V} is $n \times n$. When m is greater than or equal to n , \mathbf{U} consists of the orthonormal eigenvectors of $\mathbf{X}\mathbf{X}^t$, and \mathbf{V} consists of the orthonormal eigenvectors of $\mathbf{X}^t\mathbf{X}$. \mathbf{Q} contains the square roots of the eigenvalues of $\mathbf{X}\mathbf{X}^t$ and $\mathbf{X}^t\mathbf{X}$, except for some zeros. If m is less than n , a corresponding decomposition is performed in which \mathbf{U} and \mathbf{V} switch roles: $\mathbf{X} = \mathbf{U} \text{diag}(\mathbf{Q}) \mathbf{V}^t$, but $\mathbf{U}^t \mathbf{U} = \mathbf{U} \mathbf{U}^t = \mathbf{V}^t \mathbf{V} = \mathbf{I}_m$.

DISCRETE FOURIER TRANSFORMATION

Fourier transformation was introduced by the French mathematician and physicist Joseph Fourier in the early nineteenth century. It decomposes a signal into its frequencies. Any functions of a variable, whether continuous or discontinuous, can be expanded in a series of sines of multiples of that variable; the result has been widely used in analysis. For a sequence $\{X_t\}$, the Fourier transformation can be expressed as a linear combination of the orthogonal trigonometric functions given in the system

$$\{\sin(2\pi kt/n), \cos(2\pi kt/n) : k = 0, 1, 2, \dots, [n/2]\}$$

where $[x]$ is the greatest integer and less than or equal to x .

Therefore,

$$X_t = \sum_{k=0}^{[n/2]} [a_k \cos(2\pi kt/n) + b_k \sin(2\pi kt/n)], t = 1, 2, \dots, n$$

This is called the *Fourier series* of the sequence X_t , and a_k and b_k are called *Fourier coefficients*. You can also write the Fourier series of X_t as

$$X_t = \sum_{k=-(n-1)/2}^{(n-1)/2} c_k e^{iw_k t}, \text{ if } n \text{ is odd}$$

or

$$X_t = \sum_{k=-n/2+1}^{n/2} c_k e^{iw_k t}, \text{ if } n \text{ is even}$$

where $w_k = 2\pi/n, k = 0, 1, \dots, [n/2]$, and the Fourier coefficients c_k are given by

$$c_k = \frac{1}{n} \sum_{t=1}^n Z_t e^{-iw_k t}$$

Thus the Fourier coefficients a_k, b_k , and c_k are related as

$$a_0 = c_0$$

$$c_k = \frac{a_k - ib_k}{2} \text{ and } c_{-k} = \frac{a_k + ib_k}{2}$$

More details can be found in Chapter 10 of Wei (1990).

Agrawal, Faloutsos, and Swami (1993) and Agrawal et al. (1995) first introduced the discrete Fourier transformation for dimension reduction of time series. In the TS Dimension Reduction node, the fast Fourier transform (FFT) is used; it is a version of the discrete Fourier transform. The TS Dimension Reduction node returns the first $[d/2]$ pairs of (a_k, b_k) . Suppose you want to have four dimensions; then it returns (a_1, b_1) and (a_2, b_2) . If you want to include the mean coefficient $(a_0 = c_0)$, the final number of dimensions is seven and the order of new variables in the reduced data set is a_0, a_1, b_1, a_2, b_2 . In the FFT implementation of the TS Dimension Reduction node, if n is a power of 2, a fast Fourier transform is used (Singleton 1969); otherwise, a chirp z-transform algorithm is used (Monro and Branch 1976).

DISCRETE WAVELET TRANSFORMATION

Wavelet transformations are widely used in many fields, such as signal processing and image analysis. They are very useful for compressing digital files and reducing image or signal noise. They also have a reversible characteristic, enabling the original series to be recovered easily after the transformation. As a discrete wavelet transformation, a simple Haar transform is used in the TS Dimension Reduction node, consisting of the pairs

$$[h_0 = 2^{-1/2}, h_1 = 2^{-1/2}] \text{ and } [g_0 = 2^{-1/2}, g_1 = -2^{-1/2}]$$

for wavelet scale coefficients and wavelet detail coefficients respectively. For example, if you have a vector $x = [2, 4, -2, 2]$, then you obtain scale coefficients and detail coefficients as follows:

Resolution Level	Scale Coefficients	Detail Coefficients
2 (original data)	$[s_{2,1} = 2, s_{2,2} = 4, s_{2,3} = -2, s_{2,4} = 2]$	
1	$[s_{1,1} = 3\sqrt{2}, s_{1,2} = 0]$	$[d_{1,1} = -\sqrt{2}, d_{1,2} = -2\sqrt{2}]$
0	$[s_{0,1} = 3]$	$[d_{0,1} = 3]$

In this table,

$$[s_{1,1} = h_0 \times s_{2,1} + h_1 \times s_{2,2}, s_{1,2} = h_0 \times s_{2,3} + h_1 \times s_{2,4}]$$

$$[s_{0,1} = h_0 \times s_{1,1} + h_1 \times s_{1,2}]$$

and

$$[d_{1,1} = g_0 \times s_{2,1} + g_1 \times s_{2,2}, d_{1,2} = g_0 \times s_{2,3} + g_1 \times s_{2,4}]$$

$$[d_{0,1} = g_0 \times s_{1,1} + g_1 \times s_{1,2}]$$

Now, if you have $[s_{0,1}, d_{0,1}]$, you can reconstruct the scale coefficient at the resolution level 1. In addition, if you have $[d_{1,1}, d_{1,2}]$, the original vector is fully restored. Therefore, if you want to reduce the dimension of \mathbf{X} from 4 to 2, you have a wavelet transformation vector of $[3, 3]$. Chan and Fu (1999) give more details about time series indexing by using wavelets, and Ogden (1997) provides general information about wavelet analysis.

LINE SEGMENT METHOD WITH MEAN OR SUM

Keogh and Pazzani (2000a) proposed the line segment method first with a mean statistic; they call it *piecewise aggregate approximation (PAA)*. This is the simplest dimension reduction method in time series. If you have a time dimension of size T , the line segment method divides the time dimension into d equal-size segments (or time intervals). After segmentation, you can compute the sum, mean, or other aggregation statistic of each segment. For example, suppose you have a time series that contains 12 time points, $\{1, 2, 5, 7, 8, 5, 5, 7, 8, 2, 5, 3\}$, and you want a reduced time series that contains 3 time points. In this case, you can transform the original series into a reduced series, $\{3.75, 6.25, 4.5\}$, that contains the mean statistic of each segment. Figure 9 shows the original series and the transformed series that contains three means. The biggest advantage of this method is that it is simple to understand and to implement.

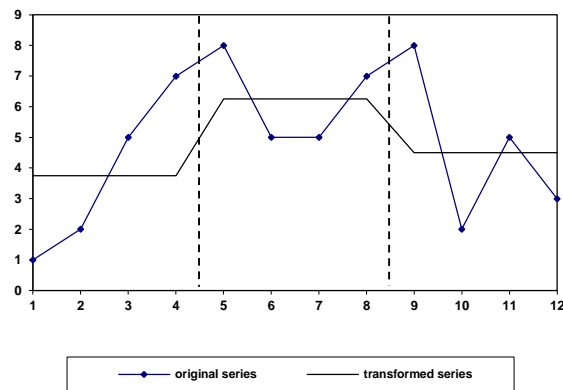


Figure 9. Line Segment Method

EVALUATION OF PERFORMANCE USING THE SYNTHETIC CONTROL CHART TIME SERIES DATA

To evaluate the performance of the feature extraction and dimension reduction methods, you use the synthetic control chart data set, which was used for similarity analysis by Alcock and Manolopoulos (1999). This data set contains 600 time series that consist of six different classes of control charts: normal, cyclic, increasing trend, decreasing trend, upward shift, and downward shift. A partial plot of the series is shown in Figure 10.

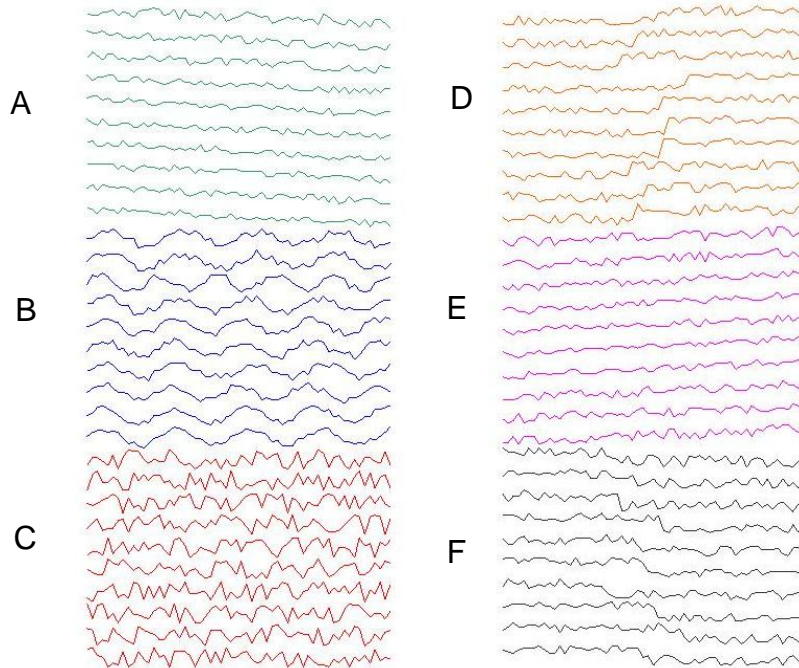


Figure 10. Synthetic Control Chart Time Series: (A) Downward Trend, (B) Cyclic, (C) Normal, (D) Upward Shift, (E) Upward Trend, (F) Downward Shift. Image courtesy of Eamonn Keogh, from the UCI KDD Archive (Hettich and Bay 1999).

EXAMPLE 1: CLUSTERING PERFORMANCE WITH TS DECOMPOSITION AND TS DIMENSION REDUCTION NODES

This example shows how to cluster the example time series by using seasonal components and dimension reduction techniques and also shows the resulting clustering performance. As a clustering tool, you use the SAS Enterprise Miner Cluster node with its default settings, except that you specify the number of clusters as 6 and the internal standardization as No. The full data set has 60 dimensions (60 time points in each series). This example shows how much benefit you can derive from using the classical time series analysis before conducting dimension reduction in time series clustering. For example, you want to separate the time series that have cyclic behaviors from the rest of the time series. You use the TS Decomposition node to extract only cyclic components; then the node suppresses all other embedded components, such as seasonal, trend, and irregular components. Figure 11 shows the diagram flow for the analysis.

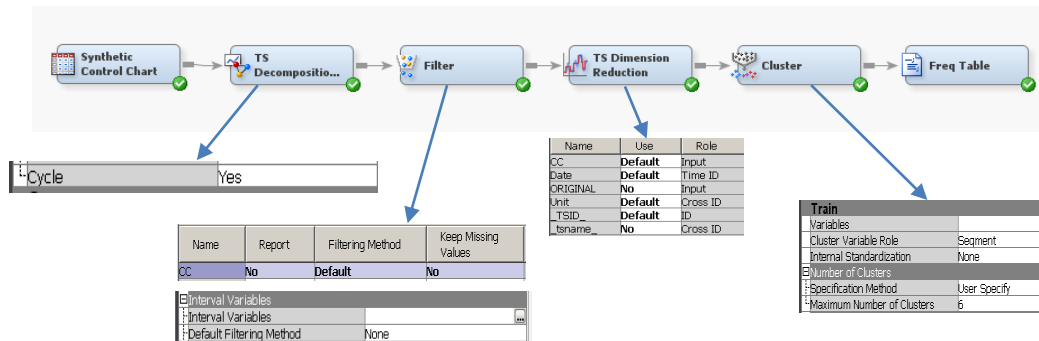


Figure 11. Time Series Clustering with Features from Classical Time Series Analysis and Dimension Reduction

The TS Decomposition node sets the export component property to Cycle. The other options in the node are set by default. You also need to use filter node to exclude some missing values in the resulted cyclic components. So you open the variable editor and set the Keep Missing Values property to No, and you set the Default Filtering Method to None. Note that CC means “Cyclic Component.” In the TS Dimension Reduction node, you use the CC variable as input and do not use the “ORIGINAL” series; you also use the wavelet transformation and the dimension of 20. The Cluster node sets the number of clusters to 6 and does not use the internal standardization. The SAS Code node reports the classified result with the original category as shown in Table 1. The table shows that 100 cyclic time series are completely separated from 500 noncyclic time series; it also shows five distinct cyclic categories.

Table 1. Clustering Results with Cyclic Components

Original Category	Classified Category	Count
Cyclic	1	21
Cyclic	2	15
Cyclic	3	22
Cyclic	5	21
Cyclic	6	21
DecreTrend	4	100
DownShift	4	100
IncreTrend	4	100
Normal	4	100
UpShift	4	100

EXAMPLE 2: PERFORMANCE EVALUATION OF DIMENSION REDUCTION TECHNIQUES

As a second example, suppose you are interested in the performance of the dimension reduction tool itself. You would like to observe whether there is any performance difference between using a reduced data set and using the original data set in the same classification tool. The reduced data set has only 10 dimensions. The dimension was reduced with a 1/6 rate. For the piecewise normalization, you use two pieces for separate normalization because the series is not too long. You directly connect two nodes (Input Data and TS Dimension Reduction nodes) to conduct the performance test. The diagram flow is shown in Figure 12.

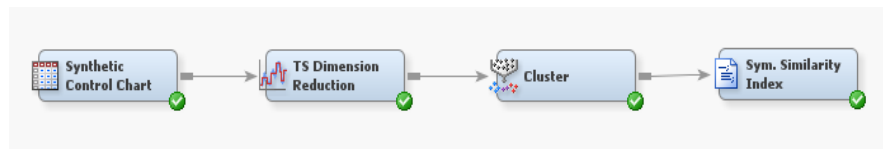


Figure 12. Diagram Flow for Performance Analysis of Dimension Reduction Techniques

In order to evaluate the results, you compute a similarity measure between the true classification of 6 types ($C_i, i = 1, 2, \dots, 6$) and the resulting clusters ($C'_j, j = 1, 2, \dots, 6$) by using the formulas (from Gavrilov et al. 2000),

$$SIM(C_i, C'_j) = 2 \frac{|C_i \cap C'_j|}{|C_i| + |C'_j|}$$

and

$$SIM(C, C') = \frac{1}{k} \sum_t \max_j [SIM(C_i, C'_j)]$$

where C_i is a true classification, C'_j is an obtained classification, and k is the number of clusters. This similarity measure returns 0 if the two clusters are completely dissimilar. If they are exactly the same, it returns 1. Because $SIM(C, C')$ is not a symmetric measure, a symmetrized version of $SIM(C, C')$ is used as follows:

$$SYMSIM(C, C') = \frac{SIM(C, C') + SIM(C', C)}{2}$$

Table 2 shows the results of $SYMSIM(C, C')$.

Table 2. Evaluation of Clustering Results

	No Normalization	Global Normalization	Piecewise Normalization (Bin = 10)
Full data	0.832	0.840	0.771
DWT	0.823	0.818	0.746
SVD	0.832	0.839	0.784
DFT	0.830	0.805	0.729
LSM	0.858	0.821	0.819
LSS	0.858	0.821	0.819

Table 2 indicates that using the full data set provides no advantage over using the reduced data sets. Ten features out of 60 dimensions produce very similar clustering results. You do not lose much information through dimension reduction. Sometimes the reduced data sets produce somewhat better results, because some noises are also reduced through the dimension reduction techniques.

CONCLUSIONS

In this paper, time series feature extraction is explained in two ways: feature extraction through classical time series and feature extraction for dimension reduction. Using these feature extraction techniques either separately or together provides a useful time series classification tool to accompany the other functions in SAS Enterprise Miner. By comparing predictions from using the full data set and using the reduced data sets, the paper shows that these dimension reduction techniques perform well. In particular, combining seasonal decomposition with the dimension reduction method shows impressive results. Keogh and Pazzani (2000b) showed that Euclidean distance measures for clustering do not perform well, and they suggest a modified version of the dynamic time warping method, which outperforms other methods. A further analysis is recommended using TS Similarity node, which provides the dynamic time warping method in SAS Enterprise Miner.

REFERENCES

Agrawal, R., Faloutsos, C., and Swami, A. 1993. "Efficient Similarity Search in Sequence Database." In *Proceedings of the Fourth Conference on Foundations of Data Organization and Algorithms*, 69–84. Lecture Notes in Computer Science 730. New York: Springer-Verlag.

- Agrawal, R., Lin, K. Sawhney, H. S., and Shim, K. 1995. "Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Database." *Proceedings of the Twenty-First VLDB Conference*, September 11–15, Zürich, Switzerland.
- Alcock, R. J. and Manolopoulos Y. 1999. "Time-Series Similarity Queries Employing a Feature-Based Approach." Paper presented at the Seventh Hellenic Conference on Informatics, Oannina, Greece.
- Chan, K. and Fu, W. 1999. "Efficient Time Series Matching by Wavelets." *Proceedings of the Fifteenth IEEE International Conference on Data Engineering*.
- Gavrilov, M., Anguelov, D., Indyk, P., and Motwani, R. 2000. "Mining the Stock Market: Which Measure Is Best?" In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 487–496.
- Keogh, E., and Pazzani, M. 2000a. "A Simple Dimensionality Reduction Technique for Fast Similarity Search in Large Time Series Databases." *Fourth Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Kyoto, Japan.
- Keogh, E., and Pazzani, M. 2000b. "Scaling Up Dynamic Time Warping for Datamining Applications." *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 285–289.
- Keogh, E., Chakrabarti, K., Pazzani, M., and Mehrotra, S. 2000. "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases." *Knowledge and Information Systems* 3(3), 263-286.
- Monro, D. M. and Branch, J. L. 1976. "Algorithm AS 117: The Chirp Discrete Fourier Transform of General Length." *Applied Statistics* 26:351–361.
- Ogden, R. T. 1997. *Essential Wavelets for Statistical Applications and Data Analysis*. Boston: Birkhäuser.
- Singleton, R. C. 1969. "An Algorithm for Computing the Mixed Radix Fast Fourier Transform." *IEEE Transactions on Audio and Electroacoustics*, AU-17:93–103.
- Wei, W. 1990. *Time Series Analysis*. Reading, MA: Addison-Wesley.

ACKNOWLEDGMENTS

The authors thank Ed Huddleston for editing this paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors:

Taiyeong Lee
E-mail: Taiyeong.Lee@sas.com

Ruiwen Zhang
E-mail: Ruiwen.Zhang@sas.com

Yongqiao Xiao
E-mail: Yongqiao.Xiao@sas.com

Jared Dean
E-mail: Jared.Dean@sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.