Paper 160-2011

Time Series Data Mining with SAS® Enterprise Miner™

Sascha Schubert and Taiyeong Lee, SAS Institute Inc., Cary, NC

ABSTRACT

Traditionally, data mining and time series analysis have been seen as separate approaches to analyzing enterprise data. However, much of the data generated by business processes is time-stamped. Time series data mining is a marriage of forecasting and traditional data mining techniques that uses time dimensions and predictive analytics to to make better business decisions. SAS has developed a collection of techniques that will be integrated into SAS® Enterprise Miner 7.1™. This paper introduces these new time series data mining techniques.

INTRODUCTION

Organizations all over the world use data mining successfully to improve their business processes, service their customers better, and optimize costs. For predictive modeling, the input data needs to be denormalized and aggregated to the entity level. For example, for a customer behavior prediction model, such as churn prediction, all input data needs to be aggregated to the customer level and presented to the modeling algorithm in a single-record vector.

Many potential predictors, however, are stored in a different format, such as time-stamped data tables, which are event based. For each event, a record is created in the table using the information related to the event. For data miners, it has been a challenge to transform time-stamped data into table formats suitable for predictive modeling. This process usually involves transforming the time-stamped data into time series data and then creating statistics as potential predictors for the predictive model. The data preparation step is one of the key elements in time series data mining.

Another area of time series data mining is pattern detection applied to the time series data directly. An example is the detection of similarities in a time series in order to identify similarities in customer behavior. For example, if unusual behavior is eminent in the time-based behavior of a particular customer, automatically detecting similar behavior in other customer data could help to uncover fraud. Once we have well-prepared data and have extracted some meaningful statistics, we can discover similar patterns among time series data or among transactional history data and cluster those into several distinct groups, such as observational clustering.

Next, we may be interested in forecasting, at least in *one-step ahead forecasting*, which is the prediction from a predictive model within each cluster. Because we handle very large amounts of data in data mining, the simplest, most effective and well-established forecasting method may be required. One of the best model groups in the forecasting field is the exponential smoothing method. In other words, data preparation, searching for similar time series, as well as clustering and forecasting within or across the segments are fundamental time series data mining processes.

This paper introduces time series data mining nodes that will be released with SAS® Enterprise Miner 7.1. The first production release of the time series data mining nodes focuses on data preparation for time series or transaction history data, similarity searches, and exponential smoothing and its implements. In subsequent releases, the time series data mining package in SAS Enterprise Miner will be expanded to other areas, such as the time-dependent regression model, time series dimension reduction, seasonal decomposition, and cross-correlation analysis.

DATA PREPARATION FOR TIME SERIES AND TRANSACTIONAL DATA

Time series data is virtually everywhere. Humans like to monitor changes over time. A random sample of 4,000 graphics from 15 of the world's newspapers published from 1974 to 1989 found that more than *75% of all graphics were time series* (Tufte, 1983). In statistics, signal processing, econometrics, and mathematical finance, a time series is a sequence of data points, typically measured at successive, uniformly spaced time intervals (Wikipedia). Uniform time intervals are equidistant time units, such as minutes, hours, days, and weeks. The time window of a time series is defined by its start value and its end value, for example from 01 Jan 2009 to 31 Dec 2009 or from 10:00 AM to 04:00 PM.

One example of a time series is the daily closing values of the Dow Jones Index. In the left panel of Figure 1, the daily values are shown over a period of one day in hourly intervals during business hours.
At the later time in the day there seems to be a long-term trend to higher values. As well, there is a shorter-term pattern of ups and downs, more pronounced at the beginning of the time interval than at the end.

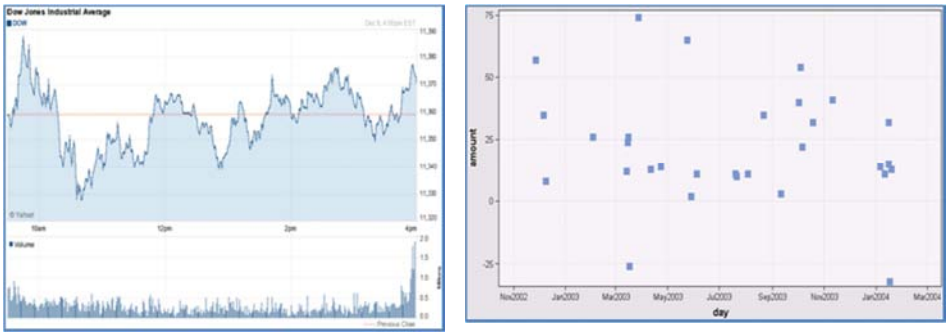


Figure 1: Time Series Data Examples

In the business world, data is often stored in a time-stamped format. This means, a record about an event of interest is created whenever this event occurs. For example, when a call is made, a call detail record is created that stores the timestamp of the call with other relevant information, such as the duration of the call, and the region called. An example of this time-stamped data plotted on a regular time-interval axis is shown in the right panel in Figure 1.

The importance of time dimension has been growing in many business applications of predictive modeling. It is intuitive that the time dimension is significant in predicting events. In marketing, the *RFM* approach has been quite successful—RFM stands for recency, frequency, and monetary value, which means that customers who have recently made a purchase are more likely to buy again. However, the format in which customer transaction data is stored in data warehouses does not lend itself to easy inclusion in predictive models.

How do we get from the raw, time-stamped, and transactional data to data that we can easily include in our predictive model? The first step is to transform the irregularly recorded time-stamped data into data measured at regular time intervals. To do this, we define a time interval that is suitable for our analysis (for example, hours, days, weeks) and accumulate the data for the chosen interval. This accumulation can result in a variety of statistics; for example, a daily total, daily average, or daily minimum and maximum. Usually, an analyst would want to use several aggregated statistics, which can lead to a substantial increase in the number of variables. Often, there are several cross-sectional variables available in a time series, such as customer regions, customer groups, or products monitored, which are called cross IDs in SAS Enterprise Miner (Figure 2). The analyst may decide which cross-sectional variables to use for data aggregation. For each category of the variables, an aggregated time series is created using the Time Series Data Preparation (TSDP) node in SAS Enterprise Miner.

As an example, this paper uses the cosmetic sales data in SAS Enterprise Miner (Figure3). This monthly sales data, collected over three years (Jan1996 through Dec1998), consists of one time ID, one target variable, and three cross-sectional variables (cross IDs: 5 products, 3 consumer groups, and 5 states), see Figure 2.

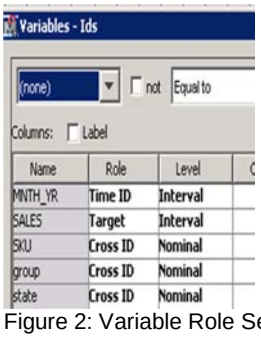


Figure 2: Variable Role Setting for Cosmetic Time Series Data

EMWS7.Ids_DATA

	Time ID	Target	CrossID: product	CrossID: group	CrossID: state
1	Jan 1, 1996	82971.0	54105	A	NC
2	Jan 1, 1996	82322.0	54105	A	GA
3	Jan 1, 1996	95834.0	54105	A	WI
4	Jan 1, 1996	98710.0	54105	A	MD
5	Jan 1, 1996	123089.0	54105	A	FL
6	Jan 1, 1996	86419.0	54105	B	NC
7	Jan 1, 1996	81668.0	54105	B	GA
8	Jan 1, 1996	75308.0	54105	B	WI
9	Jan 1, 1996	84005.0	54105	B	MD
10	Jan 1, 1996	91913.0	54105	B	FL
11	Jan 1, 1996	95748.0	54105	C	NC
12	Jan 1, 1996	99522.0	54105	C	GA
13	Jan 1, 1996	107240.0	54105	C	WI
14	Jan 1, 1996	102089.0	54105	C	MD
15	Jan 1, 1996	124963.0	54105	C	FL
16	Feb 1, 1996	83338.0	54105	A	NC
17	Feb 1, 1996	71453.0	54105	A	GA

Figure 3: Snippet of Cosmetic Sales Time Series Data

After running the TSDP node in a process flow as shown in Figure 4, you can get 75 distinct time series, which are labeled with time series IDs (TSIDs) (Figure 5). The Time Series Viewer (TS Viewer) node shows those 75 series as displayed in Figure 6.

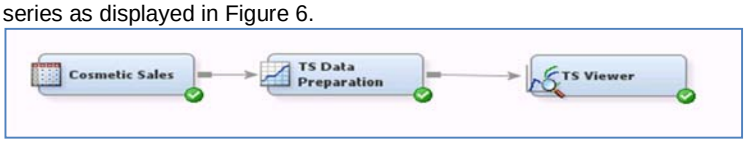


Figure4: Process Flow for Time Series Data Preparation

After running the TSDP node in a process flow as shown in Figure 4, you can get 75 distinct time series, which are labeled with TSIDs (Figure 5). The Time Series Viewer (TS Viewer) node shows those 75 series as displayed in Figure 6.

TSID Map Table

TSID	CrossID: product	CrossID: group	CrossID: state
154105	54105	A	FL
254105	54105	A	GA
354105	54105	A	MD
454105	54105	A	NC
554105	54105	A	WI
654105	54105	B	FL
754105	54105	B	GA
854105	54105	B	MD

Figure 5: TSIDs Created for Distinct Time Series

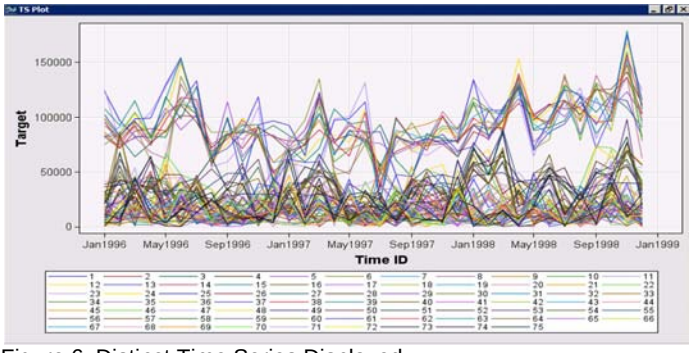


Figure 6: Distinct Time Series Displayed

Now, we can decide to aggregate the sales across all regions and groups by setting the STATE and GROUP variables to REJECTED in the Input Data Source node or by setting those two variables to No in the Use column in the TSDP node (Figure 7). The preferred accumulation method can be selected in the property sheet of the TSDP node; we will accumulate the data to monthly averages. The accumulation across time (monthly averages), regions and groups results in 5 time series that can be displayed using the TS Viewer node (Figure 8).

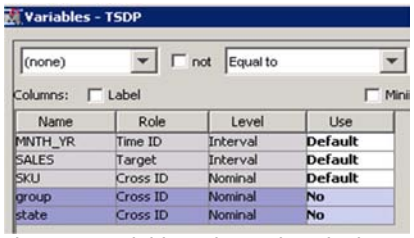


Figure 7: Variable Role Settings in the TSDP Node

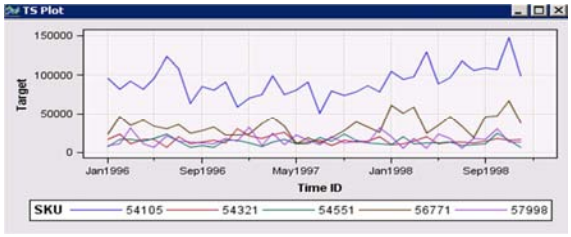


Figure 8: Display of Accumulated Time Series

In addition to the aggregation, the TSDP node provides several other techniques, such as:

- creating time series IDs and metadata
- detecting and specifying time intervals
- seasonality information
- start times and end times
- missing value replacement
- differencing
- transforming and transposing time seriesdata.

Let's look at the transpose options for clustering time series. There are two types of transpose options: By TSID and By Time ID (Figure 9). The former is useful for simarity search and time varying covariate regression models; the latter is useful for time series clustering.

Figure 2 Transpose Settings in TSDP Node

Using the TSID to transpose the input table, you can see that the result is a data set that has75 time series variables, one for each distinct value of the Cross IDs (5 states, 5 groups, and 5 SKUs)

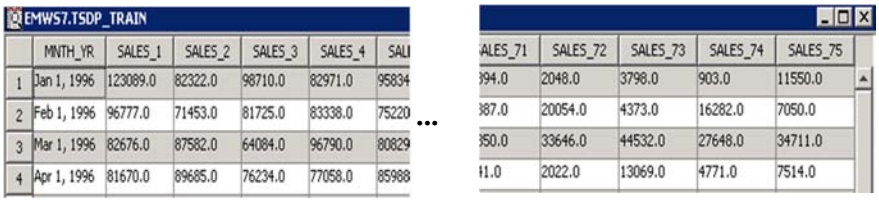
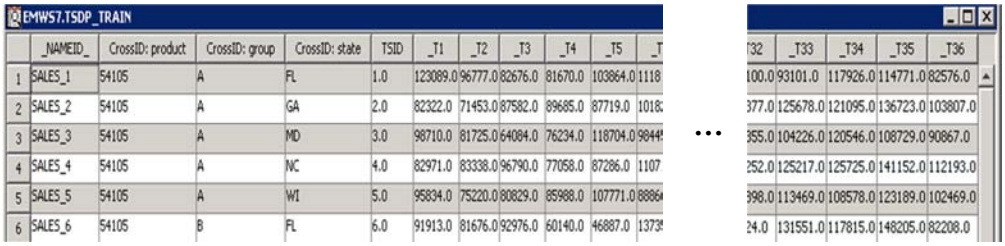


Figure 3 Time Series Data Transposed by TSID

This data set is being used to explain the capabilities of the Similarity node later in this paper. Next, for the naïve time series clustering, each time point should be converted to a coordinate variable. This can be done by transposing by Time ID (By Variable | By Time ID).



	NAMEID	CrossID: product	CrossID: group	CrossID: state	TSID	_T1	_T2	_T3	_T4	_T5	_T
1	SALES_1	54105	A	FL	1.0	123089.0	96777.0	82676.0	81670.0	103864.0	1118
2	SALES_2	54105	A	GA	2.0	82322.0	71453.0	87582.0	89685.0	87719.0	1018
3	SALES_3	54105	A	MD	3.0	98710.0	81725.0	64084.0	76234.0	118704.0	9844
4	SALES_4	54105	A	NC	4.0	82971.0	83338.0	96790.0	77058.0	87286.0	1107
5	SALES_5	54105	A	WI	5.0	95834.0	75220.0	80829.0	85988.0	107771.0	8886
6	SALES_6	54105	B	FL	6.0	91913.0	81676.0	92976.0	60140.0	46887.0	1373

_T32	_T33	_T34	_T35	_T36
100.0	93101.0	117926.0	114771.0	82576.0
377.0	125678.0	121095.0	136723.0	103807.0
355.0	104226.0	120546.0	108729.0	90867.0
252.0	125217.0	125725.0	141152.0	112193.0
398.0	113469.0	108578.0	123189.0	102469.0
24.0	131551.0	117815.0	148205.0	82208.0

Figure 11: Time Series Data Transposed by Time ID

Each time point becomes an input variable, such as _T1 for JAN1996 and _T2 for FEB19. Because we have three years' worth of monthly data, the data transposing creates 36 _T variables (Figure 11). The _T variables can be used as inputs for time series clustering. Note that when there are too many time points, some special-dimension reduction techniques are required for the time series clustering. After a TS Hierarchical Clustering (TSHC) node is connected to the transposed data and run the clustering result shows three distinct clusters selected by the Cubic Clustering Criterion (Figure 12).

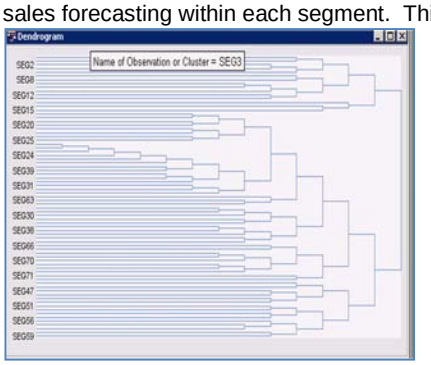
Custered Segments

TSID	_NAMEID_	CrossID: product	CrossID : group	CrossID : state	Cluster
1	SALES_1	54105	A	FL	3
2	SALES_2	54105	A	GA	3
3	SALES_3	54105	A	MD	3
⋮					
58	SALES_58	56771	C	MD	2
59	SALES_59	56771	C	NC	2
60	SALES_60	56771	C	WI	2
61	SALES_61	57998	A	FL	1
62	SALES_62	57998	A	GA	1

Figure 12: Process Flow and Result Data Set for TS Clustering

A hierarchical cluster dendrogram plot and a cluster constellation plot are shown for the TSHC result (Figure 13). The segment labels (such as SEG01, SEG12) represent pre-clusters, so each pre-segment may contain several time series or just one time series.

Instead of the TSHC, the regular clustering node in Enterprise Miner could be used. The cluster information can be merged into the original time series and we can do additional analyses for each segment, such as average sales forecasting within each segment. This is shown later in the paper.



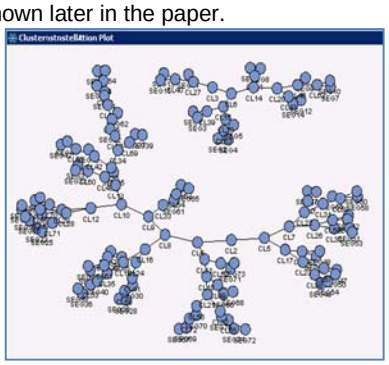


Figure 13: Dendrogram and Constellation Plot

SIMILARITY ANALYSIS

Similarity analysis can be used to compare time series and to find the series that exhibit similar characteristics over time. Finding time series with similar statistical behavior can help to make inferences about different items under observation. Similarity analysis has a number of business applications, such as the following:

- Identifying financial behavior that is similar to previously identified abusive or fraudulent behavior
- Identifying stocks or companies with similar behavior over time
- Identifying similarities in copyrighted material

The similarity analysis allows us to compare two time series, with their different lengths, using a dynamic time-warping method as well as sliding. That is, in order to account for different temporal behavior, a similarity analysis measures the distance between the input sequence and the target sequence while taking into account the ordering. Time shifting (sliding) successively shifts the input series along the time dimension to find similar patterns that might occur at different points in time. Warping refers to stretching or compressing the time dimension in order to account for slightly different time horizons in similar patterns. For example, a seasonal cycle in one stock might have a wavelength of 5 months, while a very similar seasonal cycle in a different stock might have a wavelength of 6 months. Warping would employ time stretching and would reveal that the shape of the seasonal cycle is very similar despite the slight shift. Several sources explaining the dynamic time warping method are available, but Leonard et al. (2007) and the PROC SIMILARITY document in SAS/ETS provide a detailed explanation of how the method has been implemented. The Similarity Node in SAS Enterprise Miner 7.1 uses PROC SIMILARITY to calculate the similarity measures between input time series or between input and target time series.

Similarity Search between Target Series and Input Series

The Similarity node can be used to identify the most and least similar sequences to sequences of interest. Consider the data set that we transposed by TSID in the previous section. It contains 75 distinct time series named SALES_1 to SALES_75. We set SALES_1 and SALES_10 to Target using the Metadata node, as shown in Figure 14; SALES_1 is the sales data for SKU: 54101, GROUP: A, and STATE: FL, and SALES_10 is for SKU:54101, GROUP:B, and STATE:WI. After running the Similarity node with its default settings, we found that the sales pattern in SALES_3 is most similar to the sales pattern in SALES_1, and the pattern in SALES_7 is most similar to that in SALES_10 (Figure 15).

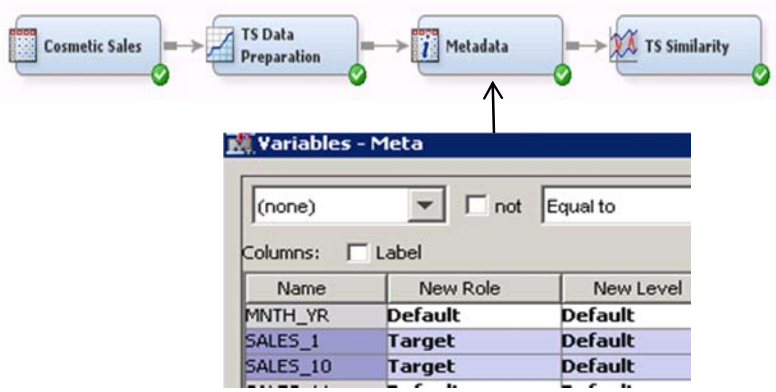


Figure 14: Metadata Settings for the Similarity Analysis

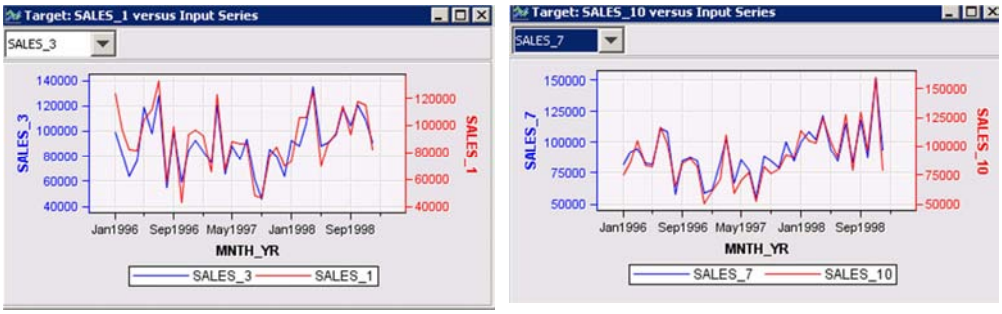


Figure 4 Most Similar Time Series

The results of the similarity node also display a matrix for the distance metrics of all input series compared to the target series.

Organizations can implement a scoring algorithm based on time series similarity that automatically detects unusual time-based behavior. For example, a bank might have uncovered fraudulent behavior in a checking account based on withdrawals and deposits. The bank can now regularly measure all other banking accounts

against the fraudulent behavior (Target Sequence) and routinely investigate the most similar accounts. The bank can also set up a threshold of required similarity and create alarms that, if the threshold is exceeded, will trigger a more thorough investigation.

Similarity Search without Specifying Any Target Series

Basically, searching for similarity without a target series is a clustering problem. If users don't specify any target information, the Similarity node will generate a distance (similarity) matrix among all input series, so the distance matrix can be fed to any clustering node. In this case, the Similarity node itself does basic hierarchical clustering. We have 75 time series that are compared against each other. The following distance matrix is obtained (Figure 16) and the dendrogram and the constellation plot of clusters are also shown in the results from the basic built-in hierarchical clustering (Figure 17).

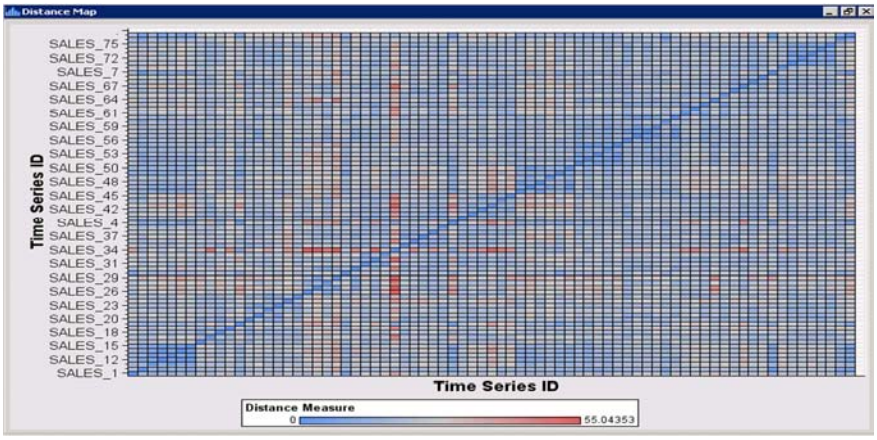


Figure 16: Similarity Matrix of 75 Time Series

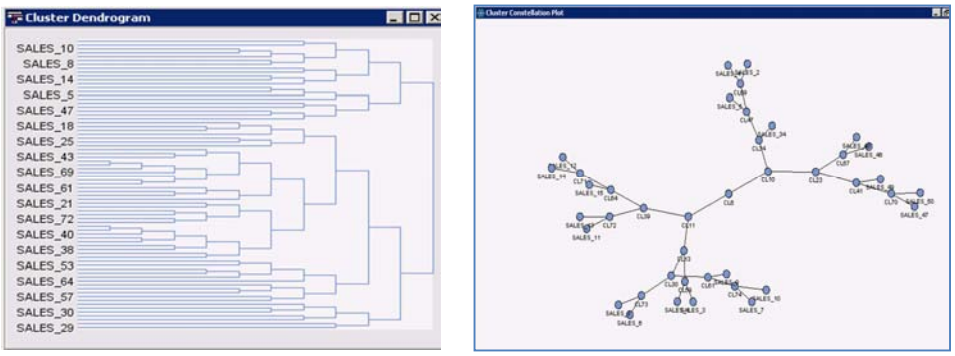


Figure 17: Dendrogram and Constellation Plot for Similarity Analysis

A more sophisticated clustering algorithm can be applied to the data after exporting the distance matrix; for example, the diagram flow could be similar to that shown in Figure 18.

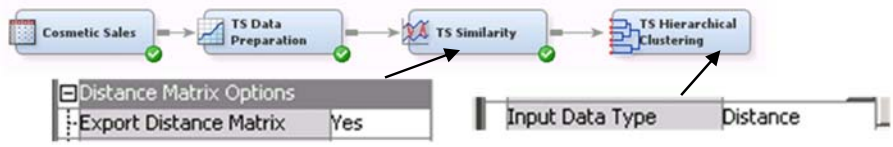


Figure 18: Combining Similarity and Cluster Analysis

By default, all other specific plots related to dynamic time warping have been suppressed because it is very expensive to generate all combinations of plots.

EXPONENTIAL SMOOTHING

Exponential smoothing is an extrapolation procedure based on a weighted moving average in which the weights decrease exponentially as data becomes older. In other words, recent observations are given relatively more weight in forecasting than the older observations. How quickly the weights decay is controlled by one or more smoothing parameters. SAS Enterprise Miner 7.1 will include a Time Series Exponential Smoothing (TS ESM) node that supports the following methods:

- Simple
- Double
- Linear
- Damped Trend
- Seasonal (Additive and Multiplicative)
- Winters Method (Additive and Multiplicative)
- Best–Best model among the models above

For more details on these methods, please see the SAS online help on the SAS/ETS procedure PROC ESM.

By default, the TS ESM node applies all exponential smoothing methods to all time series identified by the Time Series ID (which is created by the TS Data Preparation Node). Using the fit statistic Mean Square Error (MSE), the best-fitting smoothing method is selected automatically. For each series, the parameters of the best smoothing method are stored. Additionally, outliers in the time series are detected based on confidence intervals. The forecasting comparison plot and the classical forecasting band plot for each time series are shown in the node result (Figure 19). The detected outliers are shown in the forecast plots. In addition to forecasting, the node exports various data in different ways. By default, the data is exported in time series format, with or without overwritten outlier values. The data can also be exported in matrix format for subsequent time series clustering as well as in a format to support time series similarity analyses.

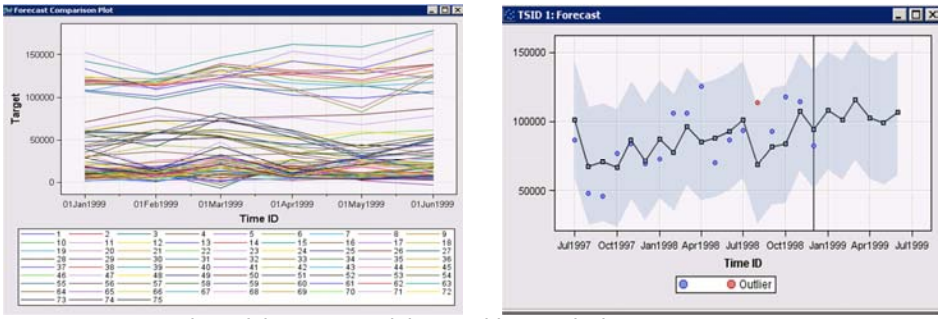


Figure 19: Forecast Plots of the Exponential Smoothing Analysis

As mentioned earlier in this paper, TS ESM node provides outlier detection using forecast confidence interval. Original values are marked as outliers when they exceed the confidence interval limits. The user can decide to leave the original outliers in the data or replace them either by the predicted values or by the missing values to apply a different missing value imputation method later. The smoothed training data below is exported for further analysis. For example, the Oct 1,1996 observation at the series of product 54105, Group A, and FL state has been replaced with the value predicted by the exponential smoothing algorithm (Figure 20).

Outliers in Export Data	
Smooth outliers	Yes
Outlier Replacement	Predicted Value

EMWS7.TSESM_TRAIN

	TSID	CrossID: product	CrossID: group	CrossID: state	Time ID	Target	Outlier
8	1.0	54105	A	FL	Aug 1, 1996	57864.0	
9	1.0	54105	A	FL	Sep 1, 1996	98903.0	
10	1.0	54105	A	FL	Oct 1, 1996	86869.99047218793	Outlier
11	1.0	54105	A	FL	Nov 1, 1996	93531.0	

Figure 20: Automated Outlier Replacement with Predicted Values

Users can leave the outlier as a missing value by setting the Outlier Replacement property to Missing; the missing value can be filled in later by a missing value imputation method (Figure 21).

EMWS7.TSESM_TRAIN

	TSID	CrossID: product	CrossID: group	CrossID: state	Time ID	Target	Outlier
8	1.0	54105	A	FL	Aug 1, 1996	57864.0	
9	1.0	54105	A	FL	Sep 1, 1996	98903.0	
10	1.0	54105	A	FL	Oct 1, 1996	.	Outlier
11	1.0	54105	A	FL	Nov 1, 1996	93531.0	

Figure 21: Automated Outlier Replacement with Missing Values

Forecast after Time Series Clustering

As mentioned briefly at the end of data preparation section, the clustered time series data can be fed to the TS ESM node to forecast the average or total sales for each segment. After clustering, the segment variable is merged with the original time series data using TS Merge node (Figure 22).

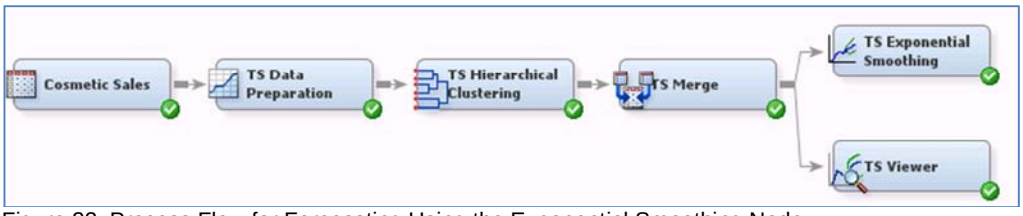


Figure 22: Process Flow for Forecasting Using the Exponential Smoothing Node

Recall that the transpose by TIME ID option was applied at TSDP node and three distinct clusters were obtained from the Hierarchical Clustering node. The TS Merge node shows the segment profiles of clusters after the merge action. In this example, the SKU cross ID is the dominant variable for the segmentation task (Figure 23).

Figure 23: Results of the Time Series Clustering

The TS Viewer node creates time series plots with clustering information (Figure 24). The first plot shows the average sales history for each segment, and the other three plots show individual plots within each segment. The first two product sales data time series are well extracted from the other three products by clustering.

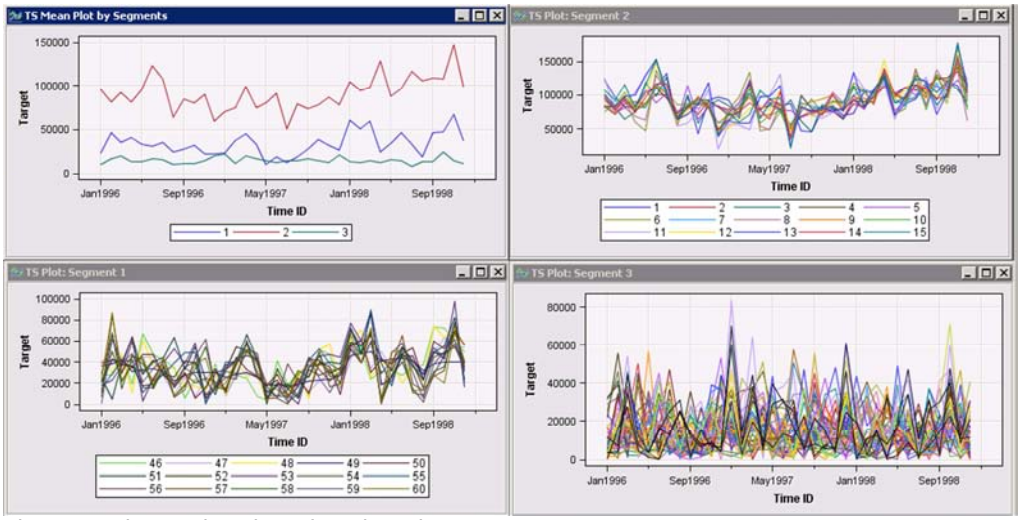
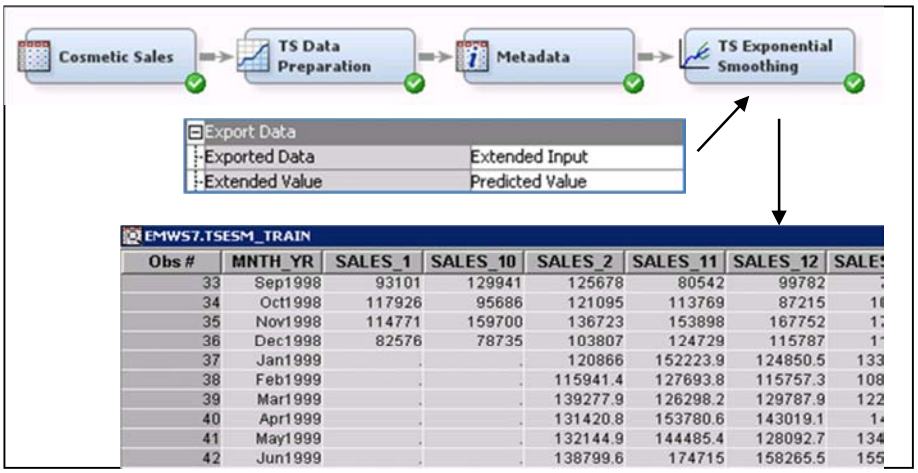


Figure 24: Time Series Plots after Clustering

Extend Time Varying Covariate to the Future Time Points in Predictive Model

When there are time-varying covariates in the predictive model, we need to extend the value of the time-varying covariate to the future time in order to score new data. Brocklebank et al. (1999) show additional details for this type of analysis. The TS ESM node provides this functionality using exponential smoothing techniques. The extended values could be predicted values, or lower or upper values of its confidence interval. To illustrate, we use the same example setting that was used earlier in the Similarity Search section, with two targets (Sales_1 and Sales_10). The TS ESM Node with the property settings of Extended Input for Exported Data shows the result: two target series have missing values at the future time points, and the other input time series have the predicted values (Figure 25).



Obs #	MNTH_YR	SALES_1	SALES_10	SALES_2	SALES_11	SALES_12	SALE
33	Sep1998	93101	129941	125678	80542	99782	
34	Oct1998	117926	95686	121095	113769	87215	1
35	Nov1998	114771	159700	136723	153898	167752	1
36	Dec1998	82576	78735	103807	124729	115787	1
37	Jan1999	.	.	120866	152223.9	124850.5	133
38	Feb1999	.	.	115941.4	127693.8	115757.3	108
39	Mar1999	.	.	139277.9	126298.2	129787.9	122
40	Apr1999	.	.	131420.8	153780.6	143019.1	14
41	May1999	.	.	132144.9	144485.4	128092.7	134
42	Jun1999	.	.	138799.6	174715	158265.5	155

Figure 25: Extending Inputs Based on Exponential Smoothing

Clustering Based on Forecast Values

Usually, time series clustering is done by detecting similar patterns in historical time series data, but sometimes a question arises as to how to cluster retail stores based on a forecasted value from the model of the historical data. This indirect time series clustering focuses on future occurrences, which are estimated by historical data. The TS ESM node provides a property to apply this kind of projection. The simplest way is to use coordinated forecast values. The forecast values at each future time point are used directly as inputs for clustering (Figure 26).

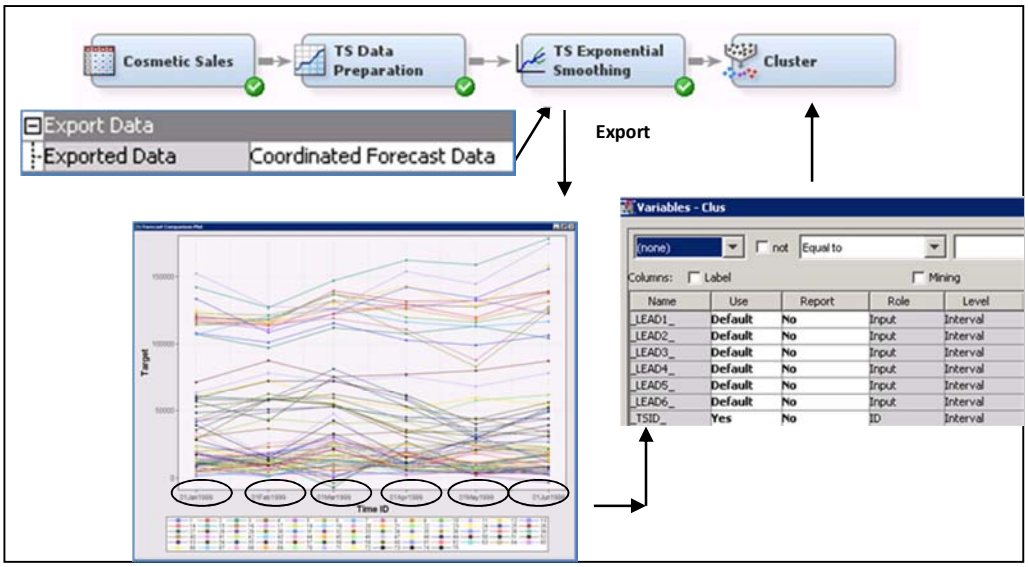


Figure 26: Using Coordinated Forecast Values for Clustering

Another way to use forecast values for clustering is to feed the forecast values to the Similarity node. Setting the property of Exported Data to Simulated Input Data creates an export data set that stores only the forecast values by TSID, so the data can be used easily in the successor node, the TS Similarity node. Based on this input, the Similarity node will generate a distance matrix for clustering analysis.

The TS ESM node also provides a unique method that incorporates the uncertainty of forecasted values used to calculate the distance matrix among forecasts using a version of Kullback-Leibler divergence. For this distance matrix, we need a fixed future time point that can be defined by setting the TS ESM property of Clustering Lead Point to 1 within forecasting leads (Figure 27). The value of 0 implements a summation over all lead points.

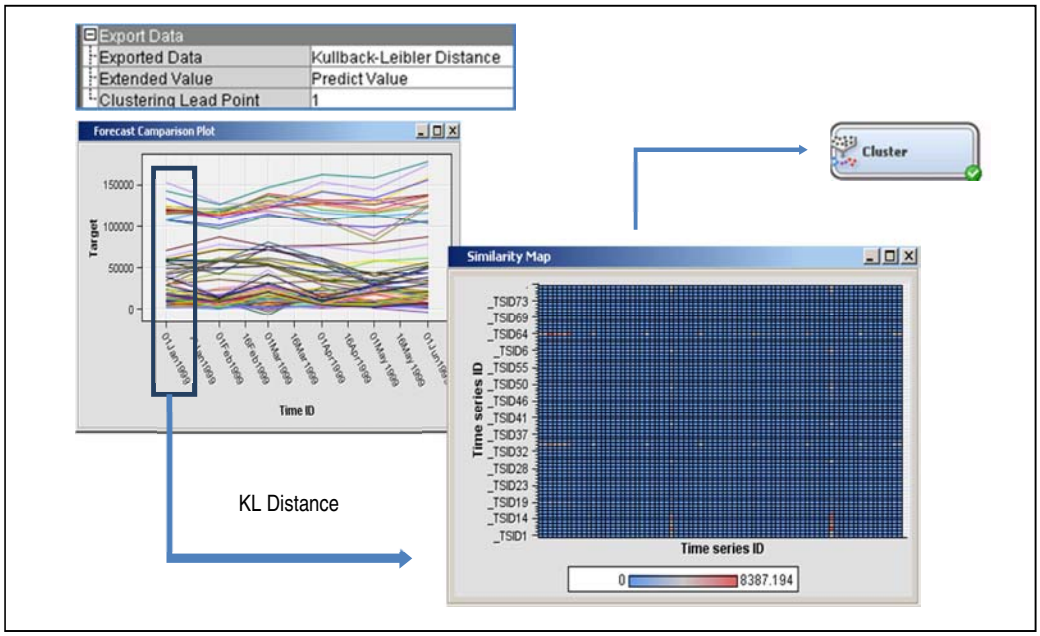


Figure 27: Applying Kullback-Leibler Distance to Similarity Analysis

Using the KL divergence and forecast clustering, the analysis can pick a group of retail stores that for the same promotion need to be considered for just the next month or even 6 months later. The details of using KL divergence will be found at Lee and Duling (2011).

In summary, the TS ESM node provides forecasting functionality in SAS Enterprise Miner as well as additional important data mining resource tools for time series and transactional data. This functionality differentiates the TS ESM node from any forecasting-only tool.

CONCLUSION

The new Time Series Data Mining nodes in SAS Enterprise Miner—TS Data Preparation Node, TS Similarity Node, and TS Exponential Smoothing Node—significantly enhance the capabilities of the data miner in the area of time series analysis and data preparation. Finding time series that exhibit similar statistical characteristics allows analysts to easily identify customer or process behaviors of interest in large volumes of time series data. With the wealth of enterprise data stored in time series, the power to integrate this data into analysis workflows will help data miners to more easily build valuable models.

REFERENCES

- Tufte, Edward R. 1983. *The Visual Display of Quantitative Information* Cheshire, CT: Graphics Press.
- Definition of "Time Series." Wikipedia. Available at http://en.wikipedia.org/wiki/Time_series.
- Leonard, Michael, Jennifer Sloan, Taiyeong Lee, and Bruce Elsheimer. 2007. "An Introduction to Similarity Analysis Using SAS®." *Proceedings of the SAS Global Forum 2007 Conference*. Cary, NC: SAS Institute Inc. Available at http://support.sas.com/rnd/app/papers/similarityanalysis.pdf.
- Brocklebank, John, Taiyeong Lee, and Michael Leonard. 1999. "Forecasting Cross-Sectional Time Series: A Data Mining Approach Using Enterprise Miner Software." *Proceedings of the 24th SAS International Users Group Conference.* Cary, NC: SAS Institute Inc.
- Lee, Taiyeong and David Duling. Manuscript in preparation. "Clustering Forecast Values Based on Kullback-Leibler Divergence."
- SAS Data Mining: http://smportal.sas.com/products/analytics/data-mining/Pages/default.aspx

ACKNOWLEDGMENTS

The authors would like to thank the following SAS employees for their valuable contributions to this paper: Udo Sglavo, EMEA Technology Practice, and Michael Leonard, SAS Research & Development.

CONTACT INFORMATION

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

Sascha Schubert
SAS Institute Inc.
E-mail: Sascha.Schubert@sas.com

Taiyeong Lee
SAS Institute Inc
E-Mail: Taiyeong.Lee@sas.com