Paper SAS212-2014

How to Separate Regular Prices from Promotional Prices?

Michael Leonard and Michele Trovero, SAS Institute Inc.

ABSTRACT

Retail price setting is influenced by two distinct factors: the regular price and the promotion price. Together, these factors determine the list price for a specific item at a specific time. These data are often reported only as a singular list price. Separating this one price into two distinct prices is critical for accurate price elasticity modeling in retail. These elasticities are then used to make sales forecasts, manage inventory, and evaluate promotions. This paper describes a new time-series feature extraction utility within SAS® Forecast Server that allows for automated separation of promotional and regular prices.

INTRODUCTION

Given a collection of time series of varying lengths, you can use time series segmentation methods to reduce the time dimension of each series by extracting important features. Time series segmentation aids in the following time series analyses:

- **Decomposing time series:** Given a single time series (original series), you can decompose the original series by determining the major (long-term) feature; you can then remove the major feature from the original series to obtain the minor (short-term) features. For example, this analysis is useful for decomposing regular and promotional prices where the original series is the sale price of a product, the major features are regular prices (long-term), and the minor features are the promotional prices (short-term).
- **Customizing discrete time intervals:** Given a single time series, you can differentiate time periods that have high and low activity in order to customize a time interval that makes the series easier to identify and easier to model for subsequent forecasting. For example, given a time series of a seasonal product has both in-season (high-activity) and off-season (low-activity) periods, it might beneficial to map the off-season period into a single time period.
- **Analyzing turning points:** Given a single time series, you can detect major change points from the time series. These change points might be useful for determining turning points in the series for subsequent turning point analysis. For example, turning point analysis is useful for forecasting the life cycle of products in the marketplace.
- **Comparing time series:** Given several time series of varying length, you can use fixed-length segmentation to extract the major features. Comparing fixed-length data vectors is far easier than comparing variable-length data vectors. For example, this analysis is useful for time series search and retrieval, ranking, clustering, and other data mining analyses that are related to time series.
- **Visualizing time series:** Given a long time series, it might be difficult to perceive which major features are associated with the time series when you view a typical (detailed) time series plot. If you extract and display only the major features, you might be able to understand the time series when you view it with less detail.

SCOPE

This paper focuses on (discrete) time series segmentation of time-stamped data that are generated by business and economic activity. In particular, this paper focuses on price series decomposition. The techniques described in this paper are less applicable to voice, image, multimedia, medical, and scientific data or to continuous-time data.

BACKGROUND

This section provides a brief theoretical background on time series segmentation analysis. It is intended to provide the analyst with motivation, orientation, and references. For an introductory discussion of these topics, see Keogh et al. (2004).

SEGMENTATION ANALYSIS

Time Indices

Let *t = 1, …, T* represent a (discrete) time index, where *T* represents the length of the historical time series.

Time Series Data

Let y_t represent a continuous-valued time series value at time index, *t*. Let $Y_T = \{y_t\}_{t=1}^{T}$ represent the time series or the historical time series vector that you want to segment. Note that Y_T can also represent a vector time series.

Segment Indices

Let *i = 1, …, N* represent a segment index, where *N* represents the number of segments. Typically, *N << T*.

Segment Series

Let x_i represent a continuous-valued, contiguous-segment series at segment index, *i*. Let $X_N = \{x_i\}_{i=1}^{N}$ represent the segment series. The segment series, $X_N = \{x_i\}_{i=1}^{N}$, more concisely describes the time series, $Y_T = \{y_t\}_{t=1}^{T}$. In other words, the segment series focuses on the most important features of the time series data. Note that X_N can also represent a vector segment series.

Time Domain versus Segment Domain

Segmentation is a mapping from the time domain to the segment domain. The *time domain* is indexed by the *time index*, *t = 1, …, T,* where *T* represents the time series length. The *segment domain* is indexed by the *segment index*, *i = 1, …, N,* where *N* represents the number of segments. Typically, *N << T*. Segmenting a time series summarizes or reduces the data that are contained in a time series while extracting the most important features.

Index Mapping

For each segment index, *i = 1, …, N,* let $[t_i, t_{i+1}) = TimeRange(i)$ represent the mapping from the segment index to a range of time indices where $1 \le \ldots t_{i-1} < t_i < t_{i+1} < \ldots \le T$. For each time index, *t = 1, …, T*, let $i = SegmentIndex(t)$ represent the mapping from a time index to a segment index. The *index mapping* segments the time indices into contiguous segments. There might be more than one time index, $[t_i, t_{i+1})$, mapped to each segment index, *i = 1, …, N*. Likewise, there might be more than one time series value mapped to each segment series value, $x_i \leftrightarrow \{y_{t_i}, \ldots, y_{(t_{i+1}-1)}\}$.

Segment Models

For each contiguous segment index, *i = 1, …, N*, a *segment model* can be defined to describe the mapping from a single segment series value to the associated range of time series values, $x_i \leftrightarrow \{y_{t_i}, \ldots, y_{(t_{i+1}-1)}\}$. Let $y_t = f_i(x_i, t : \theta_i) + \varepsilon_t$, where $t \in [t_i, t_{i+1})$, $f_i(\ )$ represents the segment model for the *i*th segment, θ_i represents the model parameter vector, and ε_t represents a disturbance term that describes uncertainty.

You can specify the segment model explicitly or the segment model can be automatically selected by using techniques similar to automatic time series modeling. Segment models can be based on summary statistics (mean, median, mode, and so on), deterministic polynomial curves (linear, quadratic, cubic, and so on), stochastic time series models (ARIMA, ESM, UCM, and so on), and other time series models (Leonard 2002 and Leonard 2004).

Fitted Segment Model, Parameter Estimates, Predictions, and Prediction Errors

For a particular segment, *i*, the associated segment model (possibly automatically selected), $y_t = f_i(x_i, t : \theta_i) + \varepsilon_t$, can be applied to the data, $x_i \leftrightarrow \{y_{t_i}, \ldots, y_{(t_{i+1}-1)}\}$. The resulting segment model parameter estimates, $\hat{\theta}_i$, can be used to predict the associated range of time series values. Let $\hat{y}_t = f_i\left(x_i, t : \hat{\theta}_i\right)$ represent the fitted model, where $\hat{y}_t$ represents the time series value predictions and $e_t = \left(y_t - \hat{y}_t\right)$ represents the model prediction errors. Let $\vec{e}_i = \left\{e_{t_i}, \ldots, e_{(t_{i+1}-1)}\right\}$ represent the *i*th segment error vector.

Segment Model Evaluation

For a particular segment, *i*, if the disturbance term, ε_t, were always zero, you could use the associated range of time series values to deterministically generate the segment series values, $x_i \leftrightarrow \left\{y_{t_i}, \ldots, y_{(t_{i+1}-1)}\right\}$, by using the equation, $y_t = f_i\left(x_i, t : \theta_i\right)$. However, this situation is not likely.

To evaluate how well the segment model performs, the error vector, $\vec{e}_i = \left\{e_{t_i}, \ldots, e_{(t_{i+1}-1)}\right\}$, can be used to create a *statistic of fit* (goodness of fit). Sum of squared errors (SSE) and absolute percentage error (APE) are examples of statistics of fit.

Let $\sigma_i = StatisticOfFit(\vec{e}_i)$ represent the specified statistic of fit that is associated with the *i*th segment model.

Merging Segments

Given an initial segmentation, *i = 1, …, N*, you can create a new segmentation by merging adjacent segments. Let *j = 1, …, (N – 1)* represent the new segmentation indexing that merges just two segments.

Suppose that the new segmentation is created by deleting the (*i+1*) segment or merging the *i*th and (*i+1*)th segments:

$$x_i \leftrightarrow \left\{y_{t_i}, \ldots, y_{(t_{i+1}-1)}\right\} \quad \text{merge} \quad x_{i+1} \leftrightarrow \left\{y_{t_{i+1}}, \ldots, y_{(t_{i+2}-1)}\right\}$$

$$x_j \leftrightarrow \left\{y_{t_i}, \ldots, y_{(t_{i+1}-1)}, y_{t_{i+1}}, \ldots, y_{(t_{i+2}-1)}\right\}$$

The goal of the segmentation algorithm is to discover the optimal segmentation based on the statistic of fit or to discover how well the segment models explain or predict the segmentations.

Cost of Merging Segments

Given an initial segmentation, *i = 1, …, N*, you can determine the cost of merging two contiguous segments to create a new segmentation: *j = 1, …, (N – 1)*.

Let σ_i and σ_{i+1} represent the statistics of fit for the *i*th and (*i+1*) segment models, respectively. Let the σ_j represent the statistics of fit for the merged segments. The cost of the merge, c_j, can be evaluated in the following ways (among others):

Absolute cost: $c_j = \sigma_j - \left(\sigma_i + \sigma_{i+1}\right)$

Relative cost: $c_j = \sigma_j / \left(\sigma_i + \sigma_{i+1}\right)$

By evaluating the cost of merging adjacent segments based on the statistics of fit, you can iteratively obtain an optimal segmentation.

Stopping Criterion

Iterative segmentation that optimizes costs often results in maximal segmentation: *N* = *T*. Maximal segmentation means that each segment consists of one time series value that can be perfectly fit. Overfitting is unsatisfactory because the purpose of segmentation is the extraction of the important features, rather than the details, of the time series. To avoid this situation, you can optionally use a stopping criterion to avoid overfitting.

Let *C* represent the *stopping criterion*. Iterative segmentation halts when the stopping criterion is satisfied: $c_i \geq C$. In other words, further segmentation would result in a small (relative or absolute) reduction in cost. By varying the stopping criterion, you can control the amount of detail that is extracted from the time series.

Segmentation Constraints

The optimal segmentation might result in one segment or in maximal segmentation: $N = 1$ or $N = T$. To avoid this situation, it might be desirable to constrain the number of allowable segments. Let m and M represent the minimum and maximum number of allowable segments, respectively. Let $\dot{m}$ and $\dot{M}$ represent the minimum percentage and maximum percentage of allowable segments based on the length of the time series, respectively. You can use these constraints to control the amount of segmentation as follows:

$$1 \le \max\left(m, \lfloor \dot{m}T \rfloor\right) \le N \le \min\left(M, \lceil \dot{M}T \rceil\right) \le T$$

When you compare the segmentation of several time series of varying length, you might want to segment each time series by using a fixed number of segments and ignoring the stopping criterion. To segment a time series to a fixed number of segments, set the minimum and maximum constraints to the same value.

For example, suppose you have two time series of differing lengths, $(T_1 \ne T_2)$, that you want to compare. Because it is difficult to compare two time series of different lengths, segmenting each time series to a fixed number of segments, N, enables you to compare the two time series more easily because the segment series have the same number of values:

$$X_{1,N} = \{x_{1,i}\}_{i=1}^{N} \leftrightarrow Y_{1,T_1} = \{y_{1,T_1}\}_{t=1}^{T_1}$$

$$X_{2,N} = \{x_{2,i}\}_{i=1}^{N} \leftrightarrow Y_{2,T_2} = \{y_{2,T_2}\}_{t=1}^{T_2}$$

Comparing fixed numbers of values $\{x_{1,i}\}_{i=1}^{N}$ and $\{x_{2,i}\}_{i=1}^{N}$ is much easier than comparing differing numbers of values $\{y_{1,T_1}\}_{t=1}^{T_1}$ and $\{y_{2,T_2}\}_{t=1}^{T_2}$.

Missing Values

If the original time series vector contains missing values, the segment boundaries are always chosen to coincide with nonmissing values. If the time series contains beginning missing values, the first segment always contains these beginning missing values. If the time series contains ending missing values, the last segment always contains these ending missing values.

Overlapping and Nonoverlapping Segmentation

The preceding description describes *nonoverlapping* segments, where a segment's ending time index is always less than the next segment's beginning time index:

$$[t_i, t_{i+1}) = TimeRange(i)$$

$$x_i \leftrightarrow \{y_{t_i}, \ldots, y_{(t_{i+1}-1)}\}$$

For nonoverlapping segments, each time index is contained in exactly one segment and the maximum number of segments is $N = 1, \ldots, T$.

It is also possible to have *overlapping segments*, where a segment's ending time index is equal to the next segment's beginning time index:

$$[t_i, t_{i+1}] = TimeRange(i)$$

$$x_i \leftrightarrow \{y_{t_i}, \ldots, y_{t_{i+1}}\}$$

For overlapping segments, each time index is contained in exactly one segment except for the first and last time index in each segment, and the maximum number of segments is $N = 1, \ldots, (T - 1)$.

Nonoverlapping segmentation is useful for segmentation that is based on summary statistics: mean, mode, median, and so on. Overlapping segmentation is useful for segmentation that is based on a curve-fitting model: linear, quadratic, cubic, and so on. The remainder of this paper describes nonoverlapping segments.

SEGMENTATION ALGORITHM

The following steps describe a bottom-up segmentation algorithm. Many of these ideas also apply to a top-down segmentation algorithm. Let *h = 1, …, (T – 1)* represent the algorithms iteration index.

1. **Initial segmentation:**

   Initially choose the segmentation indices, *i = 1, …, N*, to provide maximal segmentation: $N_0 = T$ and $t_i = i$ for nonoverlapping segments and $N_0 = T - 1$ and $t_i = i$ for overlapping segments. At this point, the *initial* segmentation is also the *current* segmentation.

2. **Evaluate the Current Segments and the Merged Segments:**

   For each current segment index, $i = 1, \ldots, N_h$:

   a. Select, fit, and evaluate a model for the current segment:

      Select a segment model: $y_t = f_i\left(x_i, t : \theta_i\right) + \varepsilon_t$

      Fit the selected segment model: $\hat{y}_t = f_i\left(x_i, t : \hat{\theta}_i\right) + e_t$

      Determine the segment errors: $\vec{e}_i = \left\{e_{t_i}, \ldots, e_{(t_{i+1}-1)}\right\}$

      Evaluate the segment model: $\sigma_i = StatisticOfFit(\vec{e}_i)$

   b. Select, fit, and evaluate a model for the pairwise merger of current and adjacent segments:

      Select a segment model: $y_t = f_j\left(x_j, t : \theta_j\right) + \varepsilon_t$

      Fit the selected segment model: $\hat{y}_t = f_j\left(x_j, t : \hat{\theta}_j\right) + e_t$

      Determine the segment errors: $\vec{e}_j = \left\{e_{t_j}, \ldots, e_{(t_{j+1}-1)}\right\}$

      Evaluate the segment model: $\sigma_j = StatisticOfFit(\vec{e}_j)$

   At this point, there are models, parameter estimates, and statistics of fit for each current segment and for the pairwise merger of the current segments.

3. **Merge segments:**

   Determine the merged segment index, *k*, that has the smallest cost increase from the initial cost to the merged cost:

   $$c_k = \min_{j} \left(\left(c_j - c_i\right) / c_j\right)$$

   Merge the segments that are associated with the *k*th merged segment index. Decrease the number of segments under consideration: $N_{h+1} = N_h - 1$.

4. **Iterate the segmentation:**

   Repeat steps 2 and 3 until the maximum segmentation constraint, $N_h \le M$, is achieved while ignoring the stopping criterion.

   Then, repeat steps 2 and 3 until the minimum segmentation constraint, $m \le N_h$, is achieved, or until the specified stopping criterion, C, is satisfied:

   $$c_i \ge C$$

When the stopping criterion is satisfied, an optimal segmentation based on the constraints is achieved. However, this segmentation might not be unique.

SEGMENTATION DECOMPOSITION ANALYSIS

Given a segmentation of a time series, $x_i \leftrightarrow \{y_{t_i}, \ldots, y_{(t_{i+1}-1)}\}$ for *i* = *1, …, N*, you can decompose the original series, y_t, into major and minor features by using the segmentation time series values predictions, $\hat{y}_t$.

Additive decomposition can be achieved by subtracting the time series values predictions (major features) from the original series to form the minor features:

$$\hat{z}_t = (y_t - \hat{y}_t) \quad \text{or} \quad y_t = (\hat{y}_t + \hat{z}_t)$$

Multiplicative decomposition can be achieved by dividing the time series values predictions (major features) from the original series to the minor features for positive time series:

$$\hat{z}_t = (y_t / \hat{y}_t) \quad \text{or} \quad y_t = \hat{y}_t \hat{z}_t$$

Other type of decompositions are also possible.

SEGMENTATION DECOMPOSITION ANALYSIS EXAMPLE

Retailers often set the price of their goods in two ways: Regular price is the price of the good when it is not being promoted; this price is usually set with their competitors in mind. Promotional price is the price of the good when it is being promoted; this price is usually set to move inventory or increase foot traffic in their stores. Usually, only the sale price is recorded (point-of sales-data), and all prices change over time. When a retailer has many stores and many items, it is difficult to determine the regular price and the promotional price for subsequent competitive and promotional price effects.

If you use only the sale price, you can use the following pricing model:

(units sold) = (sale price elasticity) x (sale price)

If you use the regular price and the promotional price, you can use the following pricing model:

(units sold) = (regular price elasticity) x (regular sale price) + (promotional price elasticity) x (promotional sale price)

The latter model more accurately captures the difference between regular price and promotional price elasticities.

You can segmentally decompose the sale price data that are collected over time to extract the regular price and promotional prices. If you use the mode of the sales price as the segment model, the segment model predictions represent the most common price, which you can assume to be the regular price (major series). You can also assume that the segment model error is the promotional price differential (minor series).

Figure 1 illustrates the daily sale price of an item that was sold at one store over three years. As you can see from the time series plot, the sale price *steps* up and down (regular price) and the sale price *spikes* up and down (promotional price). Figure 2 illustrates how using the *mode* as the segment model the segments the daily sale price. As you can see from the segmentation plot, the regular price (major series) is extracted from the sale price. Figure 3 illustrates the segmentation predictions (major series). Figure 4 illustrates the segmentation error (minor series).

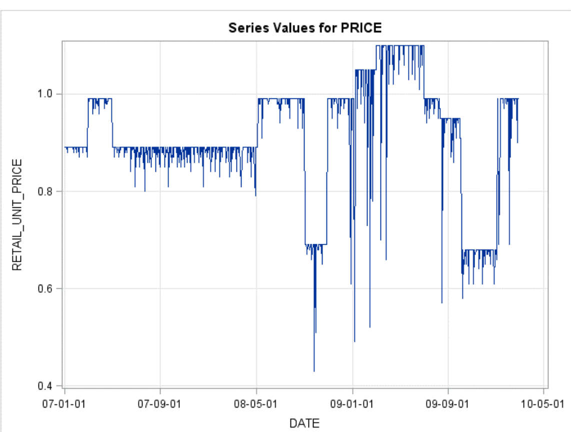


Figure 1: Series Plot for Price

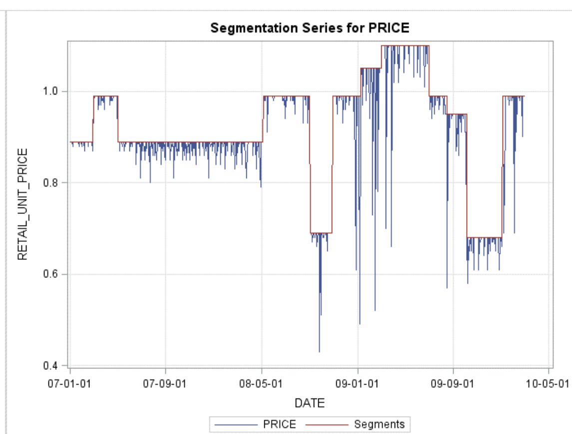
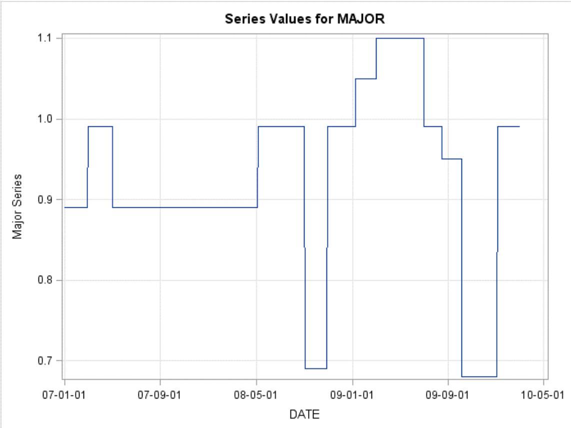


Figure 2: Segmentation Decomposition Plot



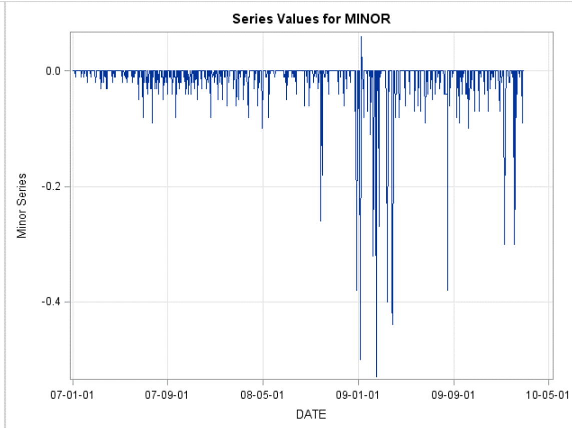


Figure 3: Major Series Plot

Figure 4: Minor Series Plot

As you can see from this segmentation analysis, the regular price (major series) and the promotional price (minor series) are extracted from the sale price.

CONCLUSION

This paper proposes techniques for large-scale time series segmentation analysis for transactional and time series data. Extracting major features from time series is useful for time series decomposition, time intervals customization, time series data mining, turning point analysis, large-scale time series visualizations, and other analyses. This paper shows how time series segmentation can be used in pricing analysis.

REFERENCES

Keogh, E., Chu, S., Hart, D., and Pazzani, M. 2004. *Segmenting Time Series: A Survey and Novel Approach*, World Scientific.

Leonard, M. J. 2002. “Large Scale Automatic Forecasting: Millions of Forecasts.” International Symposium of Forecasting.

Leonard, M. J. 2004. “Large Scale Automatic Forecasting with Calendar Events and Inputs.” International Symposium of Forecasting.

Leonard, M. J. and Wolfe, B. 2005. Mining Transactional and Time Series Data. *Proceedings of the Thirtieth Annual SAS Users Group International Conference*. Cary, NC. SAS Institute Inc. Available at http://www2.sas.com/proceedings/sugi30/080-30.pdf

Leonard, M. J. and Trovero, M.A. 2008. “A Structured Analogies Approach for Predicting the Demand of New Products.” International Symposium of Forecasting.

CONTACT INFORMATION

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

Michael Leonard
SAS Campus Drive
Cary, NC 27513
919-531-6967
Michael.Leonard@sas.com