C-TREND: A New Technique for Identifying Trends in Transactional Data

Gediminas Adomavicius
gadomavicius@csom.umn.edu
Assistant Professor

Jesse Bockstedt
jbockstedt@csom.umn.edu
Doctoral Candidate

Information and Decision Sciences
Carlson School of Management
University of Minnesota
Minneapolis, MN

Last Modified: October 16, 2006

Abstract: We introduce C-TREND, a new data visualize technique for representing trends in multi-attribute transactional data. C-TREND performs meta-analysis on standard agglomerative hierarchical clustering solutions for multiple time period data partitions to generate trend graphs. C-TREND uses nodes to represent commonly recurring transaction types in each period and edges to represent cross period relationships between the common types. In this paper, we provide an overview of the C-TREND technique, discuss the procedures for node and edge discovery from temporal data, and demonstrate an application of the technique by analyzing the change in technical characteristics of wireless networking technologies over five years.

Key Words: clustering, data and knowledge visualization, data mining, interactive data exploration and discovery, mining methods and algorithms

This is a working paper, please do not cite or distribute without the permission of the authors.
1. Introduction

In today’s information-driven world, data is captured on virtually every possible event that takes place. The research field of data mining has developed sophisticated methods for identifying patterns in data in order to provide insights to users. Identifying temporal relationships (e.g., trends) in data constitutes an important problem that is relevant in many business and academic settings, and the data mining literature has provided analytical techniques for some specialized types of temporal data, e.g., time series analysis [1, 5, 7] and sequence analysis [6, 9] techniques. However, temporal data can take many forms, most commonly being general (i.e., multiattribute) transactional data with a timestamp, for which time series or sequence analysis methods are not particularly well suited. The ability to identify trends in such general temporal data can provide significant benefits, e.g., to provide competitive advantage to a firm performing forecasts or making decisions on future investments and strategies.

In this paper, we present C-TREND, Cluster-based Temporal Representation of Event Data, a new method for discovering and visualizing trends and temporal patterns in transactional attribute-value data that builds upon standard data mining clustering techniques (Section 2). In particular, C-TREND separates data into user-defined time periods and then identifies clusters of the dominant transaction types occurring within each time period. Clusters are then compared to the clusters in adjacent time periods to identify cross-period similarities and, over many time periods, trends are identified. Trends are presented in a graph-based manner using nodes to represent clusters and edges to represent cross-time relationships. We illustrate the capabilities of C-TREND by identifying temporal patterns and trends in Wi-Fi technologies using a dataset of approximately 2400 Wi-Fi Alliance (wi-fi.org) product certifications (Section 3). We also provide conclusions at the end of the paper (Section 4).
2. The C-TREND Technique

C-TREND is designed to work with what we term *transactional attribute-value data*. Specifically, transactional attribute-value data is a general form of temporal data that consists of a collection of records (e.g., stored in a relational table) each with a time (and/or date) stamp and described by a set of attributes. Examples of this type of data include shopping cart data with a sale timestamp and numerical values representing the number of products purchased in certain categories (e.g., dairy, meat, produce) as well as product description data that includes a release date and a set of indicators representing the presence and/or quantity of specific product features (e.g., for a laptop computer: USB port, SD card slot, monitor output).

**Figure 1.** Overview of C-TREND process.

In the proposed C-TREND technique, the user provides three basic trend parameters that determine the view of data in the output graphs (see Figure 1). First, the user specifies the *trend granularity* $\Delta$ which is used to partition the initial dataset $D$ into a series of $t$ equally-sized time-period data sets $D_1, \ldots, D_t$, each of which represents a time window of size $\Delta$. Next, each data partition $D_i$ is clustered to produce a set of nodes $V_i$ for the trend graph. The user specifies $\alpha$, the *within-period trend strength* (as measured by relative cluster size), which determines a threshold for the minimum-size cluster to be included in the analysis for each time period. Nodes (clusters) that are smaller in size than this threshold are disregarded in the trend analysis. The user also specifies $\beta$, the *cross-period trend strength* (as measured by cluster similarity), which
determines a distance threshold for comparing clusters in adjacent time periods. Edges $E_{i,j}$ are generated between clusters in adjacent partitions $D_i$ and $D_{i+1}$ whenever the cross-time period cluster distances are less than or equal to the threshold. C-TREND outputs a graph of nodes, labeled with cluster sizes, and edges, labeled with their corresponding distances. The resulting graph spans across the total time period reflected in the raw data, and the paths created by edges across time periods identify trends in the data. Algorithm 1 provides the basic C-TREND procedure.

---

**Algorithm 1. C-TREND**

INPUT: D – dataset  
$\Delta$, $\alpha$, $\beta$ – trend parameters  

1  begin  
2  \{ $D_i$ \}_{i=1,t} = PARTITION(D, $\Delta$)  
3  for $i$ = 1 to $t$  
4  $V_i$ = NODES($D_i$, $\alpha$)  
5  for $i$ = 1 to $t-1$  
6  $E_{i,i+1}$ = EDGES($V_i$, $V_{i+1}$, $\beta$)  
7  end  

---

2.1 Trend Node Discovery through Clustering

To discover nodes of the trend graph, each partition is clustered to identify common naturally-occurring patterns in the data. The C-TREND subroutine NODES takes a data partition and the $\alpha$ parameter as input and outputs a set of cluster nodes for that partition. Most standard data mining clustering techniques are based on measuring distance between clusters, and there has been extensive research in data mining on clustering techniques [2, 3, 4]. C-TREND is a general technique – it can use many different standard clustering techniques (e.g., agglomerative or divisive hierarchical clustering or partition-based clustering); many cluster distance metrics,
including minimum-link (nearest neighbor), maximum-link (farthest neighbor), average-link, and the distance between cluster centers (as used in our current analysis); and many basic distance metrics between individual data points (e.g., Euclidean, Manhattan, Minkowski). Specifically, for our experiments we utilized agglomerative hierarchical clustering based on the Euclidean distance between cluster means. Agglomerative hierarchical clustering procedures start with \( n \) singleton clusters (i.e., each cluster is a single data point) and successively merge the two “closest” clusters over \( n-1 \) iterations until one comprehensive cluster is assembled.

**Figure 2** Determining Optimal Number of Clusters

The NODES procedure (see Algorithm 2) first clusters the data in the given partition, based on a predetermined clustering technique, resulting in a set of clusters (line 3). Many clustering techniques assume that the number of clusters is known ahead of time (e.g., k-means clustering) and, therefore, a common problem in cluster analysis is deciding the optimal number of clusters present in a data set. C-TREND determines the optimal number of clusters by using the so-called “gap”/“elbow” criterion [8] for comparing mean squared error (MSE) of different cluster solutions. For example, in agglomerative hierarchical clustering, \( n \) solutions are created containing 1,…, \( n \) clusters respectively. C-TREND determines the largest jump in the MSE across these solutions; a sharp elbow in the MSE plot (i.e., a significant “flattening” in the solution fitness increase) indicates an optimal number of clusters (see Figure 2). By default, C-
TREND will output data clusters based on this optimal solution; however, it can be easily modified to provide different clustering solutions based on user input. C-TREND next determines if the identified clusters are strong enough to be included in trend analysis. For each cluster $x$ discovered in data partition $P$, if the cluster size $|x| \geq \alpha |P|$, then cluster $x$ is added to the set of valid nodes that is returned for trend analysis (line 5). $\alpha |P|$ is the minimum node-size threshold, where $|P|$ is the size of the data partition passed into NODES. The $\alpha$ parameter (within-period trend strength) is between 0 and 1 and represents a percentage of the total number of data points within a data partition. For example, $\alpha = 0.02$ denotes a 2% threshold for smallest displayed cluster size. Therefore, a clustering solution for 500 data points would only include clusters of size 10 or greater. Smaller clusters would be considered spurious and omitted from the solution.

2.2 Edge Discovery and Trend Representation

Once the clustering solutions for each data partition are determined, C-TREND performs a cross-period comparison of clusters to determine similarities and identify trends. The EDGES procedure (see Algorithm 3) accepts two cluster sets $V_1, V_2$ (for adjacent-period data partitions) and the $\beta$ parameter (cross-period trend strength) as input and outputs the set of valid edges between the two data partitions.

The edge threshold $\eta$ is calculated (line 3) by taking the average of the minimum within-period cluster distances of the two partitions and adjusting it by the user specified $\beta$ parameter:

$$\eta = \beta \left( \min_{1 \leq i \leq \beta} d_{\min}(V_i) + \min_{1 \leq j \leq \beta} d_{\min}(V_j) \right) / 2 .$$

In our implementation of EDGES, the distance between two clusters $x$ and $y$ is measured as the Euclidean distance between cluster centers (means) $m_x$ and $m_y$, i.e., $d(x, y) = \| m_x - m_y \|_2$, and the
minimum within-period cluster distance is $d_{\text{min}}(V) = \min_{x,y \in V, x \neq y} d(x, y)$. The EDGES procedure is easily extensible to support other distance metrics (e.g., Manhattan) and cluster similarity comparison techniques (e.g., average-link metric).

EDGES calculates the cross-period cluster distance $d(x, y)$ for each pair of clusters $x \in V_1$, $y \in V_2$ in the two adjacent input partitions using some cluster distance metric (line 6). If $d(x, y) \leq \eta$ then the edge is added to the set $E$ (line 8), which represents valid edges between the two partitions. This approach is based on the idea that trends within each time period (as represented by different clusters) are separated from each other by at least $d_{\text{min}}(V)$ and that cross-period trend identification should be done consistently with the within-period trend identification, i.e., using comparable distance thresholds. As an example, let $\beta = 0.8$, $d_{\text{min}}(V_1)=1.4$, and $d_{\text{min}}(V_2)=1.8$. Edges will be generated for cluster pairs $(x, y)$, where $x \in V_1$ and $y \in V_2$, whenever $d(x, y) \leq 1.28$.

---

**Algorithm 2. NODES**

**INPUT:** $P$ – partition of data  
$\alpha$ – within-period trend strength  

1 begin  
2 $V = \emptyset$  
3 CLUSTERS$= \text{CL-ALGORITHM}(P)$  
4 foreach $x \in \text{CLUSTERS}$  
5 \hspace{1em} if $|x| \geq \alpha |P|$ then  
6 \hspace{2em} Generate node $v(x)$ labeled with $|x|$  
7 \hspace{2em} $V = V \cup v(x)$  
8 return $V$  
9 end
Example trend graphs generated by C-TREND are provided in Section 3.

3. Proof of Concept: Wi-Fi Technology Analysis

C-TREND is a versatile technique for identifying and representing patterns in temporal data. As mentioned earlier, the user specifies three important parameters for determining the output trend graphs: the time-period partition size $\Delta$, the within-period trend strength $\alpha$, and the cross-period trend strength $\beta$. In addition, C-TREND is highly customizable and can include multiple clustering methods and distance metrics. To demonstrate the use of C-TREND for identifying trends in transactional attribute-value data and show how modifying input parameters effects the trend graph output, we present the analysis of data on over 2,400 certifications for new wireless networking (802.11) technologies awarded by the Wi-Fi Alliance (wi-fi.org).

---

1 From Wi-Fi.org: “The Wi-Fi Alliance is a global, non-profit industry trade association with more than 200 member companies devoted to promoting the growth of wireless Local Area Networks (WLAN). Our certification programs ensure the interoperability of WLAN products from different manufacturers, with the objective of enhancing the wireless user experience.”

---
Wi-Fi Alliance certifications are awarded for ten different technology categories: access points, cellular convergence products, compact flash adapters, embedded clients, Ethernet client devices, external cards, internal cards, PDAs, USB client devices, and wireless printers. Products can be certified based on a set of standards including IEEE protocol (802.11a, b, g, d, and h), security protocol (e.g., WPA and WPA2), authentication protocol (e.g., EAP and PEAP), and quality of service (e.g., WMM). Each product certification consists of a date of certification and a set of binary attributes indicating the presence of the standards listed above.

For analysis purposes, we coded the certification data to include all standards-related attributes as well as the product type (e.g., PDA, internal card) and product category (component, device, and infrastructure) attributes. Certifications were coded into product categories based on the similarity of their product type and functionality; e.g., compact flash adapters, internal cards, external cards, and USB clients were grouped into the component category because they all act as components that provide Wi-Fi functionality to existing products, such as PCs or laptops.

Figures 3a and 3b present trend graphs for the Wi-Fi data partitioned into one-year time periods (i.e., Δ = 1 year). For each time period, a set of clusters has been identified as nodes. Each node is labeled with the size of the cluster and can be intuitively described by its center – a data record (or a vector) where each attribute value contains the mean value for this attribute across all data points in this cluster. For example, in Figure 3a the cluster labeled 82 in 2001 contains 82 data points and has a center vector (1, 0, 0, 0, 0.01, 0, 0, 0.46, 0.38, 0, 0.15, 0, 1, 0, 0, 0, 0.04, 0.04, 0, 0, 0.04, 0, 0, 0, 0, 0), which is automatically “translated” for the user to indicate that this cluster is made up of 100% components with 1% compact flash cards, 46% internal cards, 38% internal cards, and 15% USB client devices. Of these components, 100% are
802.11b-certified and 4% have WPA-personal, WPA-enterprise, and EAPTLS certifications. Based on this information, we see that all components (which are mostly cards) are clustered together at this point in the timeline. Then edges are drawn between nodes in adjacent time periods to represent similarities in clusters over time. For example, the edge labeled 0.04 in Figure 3a means that the center of the cluster labeled 56 in 2001 is only 0.04 away from the center of the cluster labeled 138 in 2002, in terms of our cluster distance metric. A distance of 0 between two clusters would mean that the cluster centers are exactly the same and, therefore, the general makeup of the clusters is essentially the same.

Comparing Figures 3a and 3b we show the effect of modifying the \( \alpha \) and \( \beta \) parameters. Figure 3a presents a C-TREND trend graph using \( \alpha = 0.02 \) and \( \beta = 0.8 \) which excludes clusters smaller than 2% of the total number of data points in a data partition and edges less than 80% of the average within-time period cluster distance. Figure 3b has more relaxed parameters, \( \alpha = 0.0 \) and \( \beta = 1.0 \) and, therefore, includes additional nodes and edges (indicated by dashed lines). The advantage of adjusting \( \alpha \) and \( \beta \) is that it provides the C-TREND user the ability to show or hide possible trends according to their strength. For example, in Figure 3b the more relaxed parameters allow the C-TREND users to uncover a new cluster in 2002 and identify a trend of 802.11b access points and printers with WPA security that was not apparent in Figure 3a. On the other hand, Figure 3b includes many clusters of size 1 with no adjoining edges. These are likely spurious events that do not provide insights on trends and can be filtered out from the graph using more restrictive parameters, as in Figure 3a.

---

2 In this paper, we do not provide such detailed information on the graphs because of the space limitations; however, this information is readily accessible to C-TREND users (e.g., by clicking on any node in the graph).
Figure 3. Wi-Fi trend graphs for one year periods ($\Delta = 1$ year).

(a) $\alpha = 0.02$ and $\beta = 0.8$

(b) $\alpha = 0.00$ and $\beta = 1.0$

In addition to adjusting the $\alpha$ and $\beta$ parameters, the user can also vary the clustering output and data attributes to customize the C-TREND trend graph output. Figure 4 provides two additional representations of the same Wi-Fi Alliance certification data. Figure 4a presents a
new cutoff in the hierarchical clustering performed by C-TREND, which allows the user to “zoom out,” i.e., to display a more aggregated view of the data by automatically combining the most similar clusters. Compared to Figure 3a, time periods 2003, 2004, and 2005 each contain only 2 clusters. Note that this is not due to the $\alpha$ parameter but instead is based on the criteria used in the hierarchical clustering. Adjusting the clustering criteria results in a more streamlined graph that identifies two major trends from 2000 to 2002 (i.e., 802.11b access points and 802.11b network cards) and then two distinct new trends (i.e., the 802.11b/g WPA security certified products and the internal cards with all standards).

**Figure 4.** Wi-Fi trend graphs for one year periods – additional views.

(a) $\alpha = 0.02, \beta = 0.8$, “zoom-out” view

(b) $\alpha = 0.02, \beta = 1.0$, “standards-only” view

Figure 4b presents a C-TREND graph of the Wi-Fi alliance certification data with product type and product category attributes removed, focusing only on standards-related data attributes. The clustering criteria for this graph are the same as those presented in Figures 3a and 3b. In this
example, the absence of product-related attributes results in the identification of clusters based solely on the certification standards and two clear trends are identified: products with and without security certification. In 2004 it is also apparent that the WMM quality-of-service standard begins to arise. In contrast, a clear start to WMM certification was not apparent Figures 3a and 3b because it was masked by the multiple clusters based on product type.

In the earlier trend graph examples, C-TREND partitioned the data based on $\Delta = 1$ year time periods. Figure 5 shows a trend graph using a different time period size, where the data from Figure 4b is partitioned based on $\Delta = 6$ months. The smaller time window provides a finer-granularity representation of the temporal trends. Compared to Figure 4b, Figure 5 clearly shows that the first products with WPA security appear in early 2001, but then seem to disappear in late 2001 and do not reappear until late 2002. It also shows some additional trends in the late 2004 and 2005 time periods. These “finer” trends were not visible in Figure 4b due to the larger time period partition.

**Figure 5.** Wi-Fi trend graph for six month periods: “standards-only” view.

<table>
<thead>
<tr>
<th>Year</th>
<th>2000</th>
<th>2001</th>
<th>2002</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
</tr>
</thead>
<tbody>
<tr>
<td>20001</td>
<td>25</td>
<td>42</td>
<td>62</td>
<td>10</td>
<td>43</td>
<td>82</td>
</tr>
<tr>
<td>802.11b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20002</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.69</td>
<td>0.65</td>
<td>0.22</td>
</tr>
<tr>
<td>802.11b w/ WPA Security</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001</td>
<td>76</td>
<td>158</td>
<td>206</td>
<td>187</td>
<td>38</td>
<td>289</td>
</tr>
<tr>
<td>0.00</td>
<td>0.01</td>
<td>0.13</td>
<td>0.32</td>
<td>0.31</td>
<td>0.96</td>
<td></td>
</tr>
<tr>
<td>802.11b</td>
<td>1.73</td>
<td>1.74</td>
<td>1.69</td>
<td>1.31</td>
<td>1.42</td>
<td>1.37</td>
</tr>
<tr>
<td>802.11b/g w/ WPA Security</td>
<td>82</td>
<td>128</td>
<td>77</td>
<td>70</td>
<td>142</td>
<td></td>
</tr>
<tr>
<td>0.06</td>
<td>0.81</td>
<td>0.81</td>
<td>0.23</td>
<td>0.47</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20011</td>
<td>253</td>
<td>253</td>
<td>253</td>
<td>289</td>
<td>289</td>
<td>289</td>
</tr>
<tr>
<td>0.09</td>
<td>0.09</td>
<td>0.09</td>
<td>0.94</td>
<td>0.94</td>
<td>0.94</td>
<td></td>
</tr>
<tr>
<td>802.11b/g w/ WPA and WPA2 Security</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20012</td>
<td>87</td>
<td>87</td>
<td>87</td>
<td>87</td>
<td>87</td>
<td>87</td>
</tr>
<tr>
<td>1.42</td>
<td>1.37</td>
<td>1.42</td>
<td>1.37</td>
<td>1.42</td>
<td>1.37</td>
<td></td>
</tr>
</tbody>
</table>

$\Delta = 6$ months, without product attributes

$\alpha = 0.02$, $\beta = 1.0$

4. Conclusions

By harnessing computational techniques of data mining, we have developed C-TREND, a novel method to uncover, analyze, and visualize trends in data. The proposed technique is
versatile and customizable and gives significant data representation power to the user – domain experts have the ability to adjust parameters and clustering mechanisms to fine-tune trend graphs and adjust the levels of granularity. In addition, C-TREND is a general technique that can be applied in a wide variety of domains. For example, as illustrated by Wi-Fi technology evolution trends in the paper, C-TREND graphs can provide support for firms performing historical analysis or making decisions for future directions based on historical trends.

References