TreeVersity: Interactive Visualizations for Comparing Hierarchical Datasets

John Alexis Guerra-Gómez
University of Maryland, College Park
CATT Laboratory & Human Computer Interaction Lab
301.405.0722
jguerrag@cs.umd.edu

Audra Buck-Coleman
University of Maryland, College Park
Department of Art
301.405.1458
buckcol3@umd.edu

Michael L. Pack
University of Maryland, College Park
Center for Advanced Transportation Technology Laboratory
301.405.0722
packml@umd.edu

Catherine Plaisant
University of Maryland, College Park
Human Computer Interaction Lab
301.405.6707
plaisant@cs.umd.edu

Ben Shneiderman
University of Maryland, College Park
Human Computer Interaction Lab
Department of Computer Science
301.405.2680
ben@cs.umd.edu

Corresponding Author: John Alexis Guerra-Gómez

Word Count: 4,999 + 10 figures = 7,999 words
Abstract

Transportation datasets of all kinds are hierarchical in nature, and therefore easily represented in tree structures. Visually comparing the changes in these hierarchical datasets over time can be extremely difficult, even when the trees are relatively small. TreeVersity is a novel interactive visualization that allows users to detect and analyze changes in the structure and value of various components of the tree. TreeVersity uses dual comparison techniques (side-by-side and explicit differences) coupled with a tabular representation to help users understand and explore changes to the tree over time. It uses carefully selected color palettes to show positive/negative, absolute and relative value changes; and glyphs that preattentively show these changes. This paper illustrates the value of this visualization technique through various short case studies using Federal transportation budget data, airline maintenance budget data and transportation publication data. Also it hypothesizes about TreeVersity’s relevance for future congestion trend analysis. Video demonstrations of this research can be viewed at [http://tinyurl.com/treeversity](http://tinyurl.com/treeversity)

Figure 1: TreeVersity comparison interface. On the top are the two original trees being compared (budgets for 2011 and 2012). At the bottom the DiffTree shows the amount of change for each node. The glyph called ”the bullet” points up to denote increase.
Introduction
Hierarchies like those shown in Figure 2↓, help us organize and understand information. Many have researched visualizing, navigating and analyzing tree structures. Techniques such as node link representations [37], TreeMaps [25], Radial representations [8] and Icicle trees [26] are now often used in scientific and non-scientific publications. However, visualizing just a single tree representing a snapshot in time has limitations. Significantly less research has been conducted on how to compare tree structures that change over time.

![Figure 2: Example datasets viewed as hierarchies](image)

Users might find themselves asking:

- Where are the significant gains and losses in a complex budget proposal?
- How and where has congestion changed both nationally and locally?
- How have airline maintenance budgets shifted from year to year, and have those changes affected safety?

The answers to these questions could be visualized though the identification of change on each node—the individual elements of the tree, each having a type (e.g. State), a name (e.g. Maryland), and a value (e.g. a budget of US$200 Million dollars for the State). These changes can be of two types: topological differences (e.g. what nodes appear, disappear or move), and node attribute value differences (increases and decreases). Most related work has focused on one or the other type of change, but not both.

Despite the substantial work on topological differences between trees, none of these solutions addresses the problem of comparing changes in node values. Thus, we propose TreeVersity, a
novel tree comparison tool able to address differences in both node values and changes in topology.

In this paper, we extend our work described first from the design perspective in [18]. We present it now from an Information Visualization point of view to illustrate its benefits as a data analysis tool. Compared to our previous work we present significant improvements such as new filtering techniques (DiffScatterPlot see Section 3.4.1↓), animated transitions to reuse screen space, colored table nodes for easy ranking and matching, and support for moved nodes. The contributions of our work include:

- Design and implementation of TreeVersity, an interactive information visualization tool for comparing trees by looking at changes both on topology and node value differences (Tree comparison Types 0,1,2,3,4 see section 1.1↓)

- Application of the tool using three different data domains, Federal transportation budget data, airline maintenance budgets and transportation publication data.

- Design and implementation of the Bullet, a visualization glyph that allows the representation of five dimensions of tree node change, including direction of the change (positive, negative or neutral), absolute differences, percentage changes, change relative to other nodes, and topological differences (created, removed and moved nodes).

The rest of the paper is organized as follows: First we define the types of tree comparisons addressed by TreeVersity. Then, we present TreeVersity and offer detailed explanations of its different components. Next we present three brief transportation hierarchical datasets from which important information can be gleaned through the use of TreeVersity. Finally we describe some conclusions of our work.

**Tree Comparison Types**

Much work has been done on visualizing [25, 26, 27, 38] and exploring [37, 19, 6] single tree structures; however, the problem of comparing two trees is significantly harder. We have identified and classified the following types of tree comparison (Figure 3↓):

- **Type 0**: Topological differences between two trees where the nodes only contain a label. Example: Finding created, moved and removed topics between two versions of the TRB research publications hierarchical categorization system, without looking for number of papers.
• **Type 1**: Positive and negative changes in leaf node values with aggregated values in the interior nodes (i.e. trees that can be visualized with a treemap [25]) and no changes in topology. Example: Comparing the change on the number of workers on the traffic agencies of the country, grouped by Agencies, Counties and Regions, given that no Agencies are created or removed.

• **Type 2**: Positive and negative changes in leaves and interior node values with no changes in topology. Example: Comparing the salaries in the Department of Transportation's organizational chart between two years, when no reorganization has occurred.

• **Type 3**: Positive and negative changes in leaf node values with aggregated values in the interior nodes and with changes in topology. Example: Again finding changes in the TRB research publications hierarchical categorization system, but looking at changes in the number of papers published by topic and at created, removed nodes.

• **Type 4**: Positive and negative changes in leaves and interior node values, with changes in topology. Example: Comparing the number of TRB’s website visits between two months using the file hierarchy as a natural organization. Some pages might be created or removed, and each page in the hierarchy has an independent number of visits.

---

**Node Value Changes**

<table>
<thead>
<tr>
<th>NO</th>
<th>YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leaf nodes only</td>
<td>All the nodes</td>
</tr>
<tr>
<td>Type 1: Values on the leaves only, same topology</td>
<td>Type 2: Values on all the nodes, same topology</td>
</tr>
<tr>
<td>Type 3: Values on the leaves, different topology</td>
<td>Type 4: Values on all the nodes, different topology</td>
</tr>
</tbody>
</table>

---

**Topology Changes**

<table>
<thead>
<tr>
<th>NO</th>
<th>YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 0: Different topology, no values</td>
<td>Type 3: Values on the leaves, different topology</td>
</tr>
<tr>
<td>Type 4: Values on all the nodes, different topology</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: Types of tree comparison problems. Current literature has addressed Types 0 and 1, with only one attempt at Type 3 [46]. TreeVersity supports all five cases, with emphasis on Types 1-4, the ones that include node value changes.
Dimensions of node changes

According to related work and our own research, analysts that want to perform these types of tree comparisons want to be able to find and understand the following dimensions of tree node changes:

- **Direction of change**: positive, negative or neutral (no change).
- **Absolute change**: the actual amount of change, e.g. the Department of Transportation budget will be decreased by 5.99 billion dollars between 2012 and 2013.
- **Percentage change**: the absolute change with respect to the original value, such as the cut in the Department of Transportation represents a 1.32% decrease with respect to its budget in 2012.
- **Relative change**: how does a node change compare to the changes of other nodes in the tree, e.g. The cut in the Department of Transportation ($-5.99 Billion, -1.32%) will be considerably smaller than the expected decrease in the Department of Labor’s budget ($-52.66 Billion, -29.84%).
- **Topology change**: Nodes that are created, removed, or moved. e.g. The Bureau of Engraving and Printing ($140 million dollars) is scheduled to be removed from the Department of Treasury on 2013.

Researchers have proposed a significant number of solutions for comparing trees on topology (Type 0) [9, 37, 11, 13, 16, 42, 1, 23, 2, 32, 34, 36, 15, 31, 7, 28, 14, 17, 5] or for visualizing changes in node values with aggregated values in the interior nodes (Type 1). Only one project [46] has attempted combining both types of differences at the same time (Type 3). For a more comprehensive description of the related work please refer to [47].

**TreeVersity**

TreeVersity is a tree comparison tool that tackles a richer set of problems by combining a novel visualization technique, interface design with coordinated views, interaction techniques and comparison algorithms to support all five types of tree comparisons. TreeVersity user target are power users interested on finding differences on hierarchical datasets (e.g. data analysts, researchers and policy and planning officials making decisions based on data). However we envision TreeVersity also as a communication tool targeting a broader audience.

**Interface design**

TreeVersity combines juxtaposition and aggregation techniques [30] along with interconnected views (Figure 1↑). The top of TreeVersity shows a side-by-side comparison of the two original trees. Below them a third aggregated view called DiffTree shows the differences between the
original trees. The three views are interconnected: selecting one node highlights and centers the two other corresponding nodes in the other views.

TreeVersity also displays the differences between trees in a tabular representation (top left of Figure 1↑). The table lists all the nodes currently displayed, also with tightly coupled highlighting. The columns include the name of the node, level in the tree, and absolute and relative differences of each attribute. Sorting columns allows the rapid selection of nodes with extreme values (e.g. largest relative difference).

**Visualization Technique**

The compared trees seen in the top of Figure 4 use rectangular icons with color and size redundantly representing the attribute values of each node. To build the DiffTree seen in the bottom of Figure 4, TreeVersity computes the difference in all the attributes for each original node. A positive difference indicates that the value on the right is larger than the one on the left. A node present on the left but not on the right is considered a removed node and its value in the DiffTree will appear as negative, assuming the value of absent nodes as zero.

Users can filter out specific nodes by differential amounts and/or by topological characteristics (created, removed or present in both trees). The nodes are sorted according to the amount of change (absolute or relative).

The DiffTree (seen in the lower half of Figure 4) uses a novel glyph visualization—the Bullet—to represent differences between the two original trees. The Bullet glyph encodes the cardinality of the change, the amount of change, and the creating/deletion. The shape’s direction represents the cardinality of the change: down for negative and up for positive in the vertical layout and left for negative and right for positive in the horizontal layout. The bullet size represents the amount of change. Color is used to encode both the cardinality and amount of change in the nodes. Users can select from preset color palettes that are binned in five steps to ease differentiation and accommodate visual preferences. The colors in the DiffTree are deliberately different from those in the original trees because they usually need very different value ranges. Gray rectangles represent nodes where the amount of change is zero. Finally, thick black or white borders are added around the bullet to denote removed or newly created nodes (white for added nodes, black for removed). By default both size and color are redundantly encoding the absolute amount of change (e.g. the amount in dollars in the case of a budget), but users can switch to relative change (i.e. percent change), or assign color and size to different characterization of the changes.
Filtering

Users can filter the nodes by topological change, by range of values, and by maximum depth. Topological change allows users to see only the nodes that were created, or removed, or that are present on both trees. With the filter-by-node-variables range, users can keep visible only nodes whose values fall within a specific range, using an absolute or relative amounts of change. Finally, the filter by maximum depth hides all the nodes that are deeper than a specified maximum depth. Another way of filtering uses a visualization called the DiffScatterPlot, seen in the callout of Figure 5, which lays out in a scatter-plot the summary of all the changes in the tree distributing dots according to their absolute amount change on the y-axis, and to their percentage of change in the x-axis. Users drag a bounding rectangle with the mouse to select nodes of interest. This technique is especially useful for selecting outliers.
After the filtering operation, any nodes that do not fit the new selection criteria will be hidden from all the views (including the table). Through animation, the empty space is reclaimed for the remaining nodes, making better use of screen.

**Overview**

All three visualizations offer panning and zooming options for navigation. When analyzing trees with thousands of nodes, a zoomed out (macro) view of the whole tree can produce a cluttered mass of nodes. To reduce clutter, TreeVersity distributes the distance between the layers of nodes to fit the screen. This option is especially useful to understand the structure of the compared trees and of the DiffTree.

**Navigation**

Users can focus on a subtree comparison. This is done by double clicking on the root node of the subtree of interest. After navigating into a subtree all the views will be updated to display only the nodes on it; this is particularly useful in de-cluttering the screen. A navigation panel records navigated nodes and allows users to return to a previously navigated state.

**Architecture**

TreeVersity was developed using Java 1.6 and the great Prefuse [48] visualization toolkit using a machine with an Intel Core i7 processor with 4GB of RAM memory using GNU/Linux, but has been successfully tested in computers with smaller specifications and with different operative systems (Mac OSX, and Windows 7). In terms of data size, TreeVersity was tested with good response times (no noticeable visual delays) comparing two trees of up to 8,000 nodes each, up to 10 levels deep, with a fan-out of less than 100 children and less than 40% of topological changes. Importers were created for CSV and XML datasets.

**Use Case 1: Airline Maintenance Budgets**

TreeVersity was used to analyze the changes in maintenance budgets of different airlines that operate in the US. The dataset, provided by the U.S. Dept. of Transportation, contained the reported amount of money spent by the airlines for maintenance by quarters and years, and included attributes such as the regions where the airlines operate and their net incomes. Two hierarchies where built at the request of the DOT analysts: first grouping the airlines by region of operation (A: Atlantic, L: Latin America, D: Domestic, P: Pacific, I: International) and then by the carriers. Then the tree was flipped. A total of 67 carriers were compared between 2011 and 2010.
The analysts started by asking which airlines changed their budget the most when grouped by regions, so using the DiffScatterPlot, TreeVersity filtered those changing the most, both in absolute and in percentage differences. After the filter, only 8 out of the 67 carriers remained (see Figure 5). Looking at these airlines, it was immediately apparent that PSA and Compass Airlines were big outliers in their percentage changes, with +305.86% and +230.96% respectively. The analysts realized that both companies had been involved in mergers which would be a likely explanation for the big increase. One DOT analyst commented that "It’s great that we could identify this airlines..." and "... if it weren’t for this visualization we wouldn't have noticed this". They also complimentsed the aesthetic quality of the framework’s design.

DOT analysts also noted that Delta and Southwest Airlines presented significant absolute increases, and that American Airlines was the biggest decreasing carrier. They then noted that it would be beneficial to compare the maintenance budgets by aircraft type to see how expenditures for Boeing 747 and 767 compared to Airbus or other manufacturers. A Multi-dimensional In-depth Long-term Case Study [40] is being planned with the DOT staff to measure insight development using TreeVersity in their datasets over a longer period of time.

Figure 5: Airlines that change their maintenance budget the most between 2011 and 2010 by region. Airlines were filtered by those that incremented their budgets in more than $27,000 or more than 200% or that reduced their budgets in more than $13,500. The budgets are grouped by regions, the nodes at the first level in the tree. D stands for Domestic and A for Atlantic. The values in the regions represent the average amount of change in all the airlines in that region. The root node shows the average overall.
Application Case 2: Change in passengers flying from US Airports

A second case study was performed on a dataset containing the total number of enplanements (number of passengers boarding planes) in each U.S. Airport, by state and by month. The data provided by the Bureau of Transportation Statistics contained 65,535 records from 1990 to 2003. The hierarchies for each year were grouped by Month, State, and Airport.

Figure 6 (top) shows the DiffTree of the year 2000 compared to 2001 by Month and then by State. The nodes in the tree are sorted by the absolute amount of change, and color is used for the change in percentage with respect to 2000. The four dark subtrees on the left show the significant decrease on the number of enplanements that occurred from September to December due to the September 11th attacks and subsequent changes in flight regulations, while the months prior to the attack like January and August showed significant increases in enplanements.

Performing a more in-depth analysis, Figure 6 (middle) shows the changes on the enplanements by State and then by Airport for the month of September only. The prominent yellow to red colors shows that decreases were reported by all States in almost all the Airports; however, some green nodes in the visualization reveal that even after 9/11, some Airports actually reported increases in their number of passengers boarding. Figure 6 (bottom) shows all the airports that reported increases over 300 passengers, with Minot (MOT) in North Dakota reporting on September 2001 an increase of more than 442% on the number of enplanements compared to the same month on 2000.

Application Case 3: National Transportation Library (NTL) Publication Data

A third case study was developed analyzing NTL’s publications database including 52,135 papers published between 1968 and 2010. Initial explorations revealed that the number of papers published fluctuates significantly between consecutive years (The changes between years from 2001 to 2010 were +224, -621, +737, +289, -864, +1,724, -61, -529, +271, +1,379). In order to make a more representative comparison two periods of five years, 2001 to 2005 and 2005 to 2010 were chosen. The compared datasets included 5,566 papers published for the
first five years and 10716 for the five years after. 962 papers didn’t include any author information, so, they appear as null authors in the visualizations. The papers were grouped by subject area and then by author. Only authors that published more than five papers per time period were considered to reduce the number of nodes on the trees that were in the order of 84,112 without the filtering.

Figure 6: Top: Change in number of passengers by month and State between 2001 and 2000, sorted by Absolute Change. Middle: California, Texas, Illinois, Florida and New York saw the biggest absolute decreases. Most airports decreased, but there are some showing increases (green edges). Bottom: Airports increasing in more than 300 passengers in September 2001 compared to September 2000. Airports like MOT in North Dakota, didn’t seem affected by 09/11, they reported an increase of 442%.
The overview of the publications subjects and authors on Figure 10, show that most of the subjects presented increases in the second part of the decade, especially with respect to common subjects like "Highways". However, some subjects like Geotechnology showed decreases. While no subjects disappeared, there were four new topics, some of which have shown significant gains like "I15: Environment" with a total increase of 662 papers. Looking at the change on the level of the Authors by Subject, Figures 7 8 9 show the authors that decreased, increased and maintained their rate of production. For sake of maintaining anonymity, these author names have been covered up in these graphics.
Figure 8: Three DiffTrees showing subject area changes (Top), authors that are showing decreases in publications (middle), and authors that are showing significant increases in NTL publications (bottom).
Conclusions

Comparing trees is very challenging—especially when trees are large and complex. TreeVersity allows users to easily identify differences between trees, see patterns and spot exceptions. Three case studies analyzing the US airline maintenance budgets, US airport enplanements, and National Transportation Library publications data were presented—illustrating just a few of TreeVersity’s features.

Ongoing research with TreeVersity includes further simplifying the user interface with respect to filtering options, enhancing labeling of nodes especially when there are significant numbers of nodes on a single screen, and creating automated reports that guide users through the most significant differences between the trees. Researchers are also studying the node limits of tree comparisons to determine how much information can be both displayed on the screen and comprehended by a user. Lastly, researchers are exploring the best way to allow the simultaneous comparison of more than two trees or the aggregation of multiple years of trees.

Researchers are now preparing a complex national dataset of hundreds of thousands of miles of congestion and bottleneck data for use in TreeVersity. It is expected that a hierarchical set of congestion measures for every single segment of every single road in every single state for the United States would work well in TreeVersity, and allow national analysts to better understand quarterly, yearly, or decade long changes and trends in congestion while better prioritizing national transportation project funding decisions. Given the interest of correspondent data
analysts, further applications are possible with other transportation related datasets like State DOT Transportation Program-Budget Planning and Preservation Evaluations, State MPO TIPs-Transportation Improvements Plans, Civil Infrastructure Construction Project Performance tracking, etc.

Limitations
Some of the limitations of TreeVersity that could be addressed in future work include:

- Support moved nodes and trees without identifiers.
- Better scalability (more than 10,000 nodes).
- Compare more than two trees.
- Better labeling using available space.
- Easier importing and database connectivity.

Acknowledgements
We want to thank the Fulbright International Science and Technology Scholarship and the Center for Integrated Transportation Systems Management (a Tier 1 Transportation Center at the University of Maryland) for funding this research. We also thank Patricia Hu, Amanda Wilson, Martin Akerman at the U.S. Dept. of Transportation for taking the time to work with us to applying TreeVersity in real world problems. Lastly, we thank Michael VanDaniker and Tom Jacobs at the CATT Lab for their suggestions and feedback.
References


