TreeVersity: Comparing Tree Structures by Topology and Node's Attributes Differences

John Alexis Guerra Gómez*
HCIL & Department of Computer Science
University of Maryland

Audra Buck-Coleman†
Department of Art
University of Maryland

Catherine Plaisant‡
HCIL & UMIACS
University of Maryland

Ben Shneiderman§
HCIL & Department of Computer Science
University of Maryland

ABSTRACT

It is common to classify data in hierarchies, they provide a comprehensible way of understanding big amounts of data. From budgets to organizational charts or even the stock market, trees are everywhere and people find them easy to use. However when analysts need to compare two versions of the same tree structure, or two related taxonomies, the task is not so easy. Much work has been done on this topic, but almost all of it has been restricted to either compare the trees by topology, or by the node attribute values. With this project we are proposing TreeVersity, a framework for comparing tree structures, both by structural changes and by differences in the node attributes. This paper is based on our previous work on comparing traffic agencies using LifeFlow [1, 2] and on a first prototype of TreeVersity.

Index Terms: E.1 [Data Structures]: Trees—; H.5.2 [User Interfaces]: Graphical User Interfaces (GUI)—

1 INTRODUCTION

Hierarchies help analysts understand and categorize information by describing responsibilities, aggregations, precedence or any other parent-to-child relationship. Examples like the US Federal Budget, the closing prices of a Stock Market, the tree of species, or even one season's NBA player statistics, can be viewed and organized as a hierarchy. Much research has been done to help us understand, visualize and navigate tree structures. Techniques like node link representations, treemaps [3], Radial representations [4]and even Icicle trees [5] are commonly used even in non scientific publications like newspapers and websites.

Once analysts understand one single hierarchy, the next question would be how to compare multiple trees. Examples of this type of problem are tasks like understanding what changes in a budget proposal compared to previous years, which species are grouped differently on two different taxonomies or which stock's prices have significantly changed on the market. As will be explained in section 2, this type of question has led to a rich group of research projects, that most commonly have been narrowed for specific domains. Because of this they either focus on finding topological differences (e.g. nodes that have been deleted, created or relocated), or comparing node attribute values (e.g. finding budget cuts between years).

This work proposes a tree comparison framework that addresses a richer set of problems, including: Positive and negative changes in leaf node's attribute values, with and without changes in topology, and positive and negative changes in leaves and interior node's attribute values, with and without changes in topology.

*e-mail: jguerrag@cs.umd.edu
†e-mail:buckcol3@umd.edu
‡e-mail:plaisant@cs.umd.edu
§e-mail:ben@cs.umd.edu

IEEE Symposium on Visual Analytics Science and Technology
October 23 - 28, Providence, RI, USA
978-1-4673-0014-8/11/$26.00 ©2011 IEEE

2 RELATED WORK

This section focuses on research that has been done on comparing, visualizing and analyzing multiple tree structures. There is substantial work on single tree structures, but since they are not relevant to the objective of comparison, it won't be described in this document..

Significant work has been done on finding topological differences between trees, TreeJuxtaposer [6] and TaxVis [7] are great examples of such tools. However these solutions are restricted to changes on the tree structure and not on the node attribute values. There are few projects that look into these changes, like the Contrast treemaps [8] but these are difficult to understand and focused on leaf node attributes only. Finally another way of comparing trees has been using metrics like edit distance, inclusion and alignment Bille et al. [9]provide a comprehensive survey of these.

3 TREEVERSITY

TreeVersity is being designed to support the following use cases: 1) Given two trees, compare them and explore the differences by user interaction. 2) Same as the previous case, but provide a ranking of the most significant differences and similarities, and guide the user through them. 3) Given a forest (a group of trees) and one selected tree as a base, find the most similar (and the most dissimilar) hierarchies, and display their similarities (and differences). 4) Given a forest, create an average tree that represents the most common characteristics among the trees. 5) Given a forest, find the most different trees, that is, trees that are the most different to the average.

Of those use cases, we have started to implement the first one, interactively comparing two tree structures. Our main goal is to create an interactive visualization that allows the comparison of two trees by looking at: 1) Created and removed nodes. 2) Absolute and relative differences of the node attribute values. 3) Cardinality of the differences. 4) Differences in attributes of leaf nodes only, or differences in attributes of leaf and interior nodes. 5) Amount of change compared with the other nodes on the tree (or compared with the siblings).

This prototype of TreeVersity uses a mixed approach for comparison. First it presents a connected side by side comparison of the trees, that allows synchronized navigation and identification of unique versus created/removed nodes. Second it displays an aggregated view that shows the differences between the node attribute values of the trees, we call this view the *diffTree*. For this view we have been experimenting with two different visualizations, that we call the "slope" and the "gas tank". In the next sections we describe them in more detail.

3.1 The Slope

The slope is based on a node link representation. It uses shape, color and size to represent the (absolute or relative) amount and direction of change.

An intersection of the trees is calculated, computing the differences of each equivalent node in both threes (using a given id for the matching). The shape of each node represents the amount of change on that node. A decreasing slope means that the value was

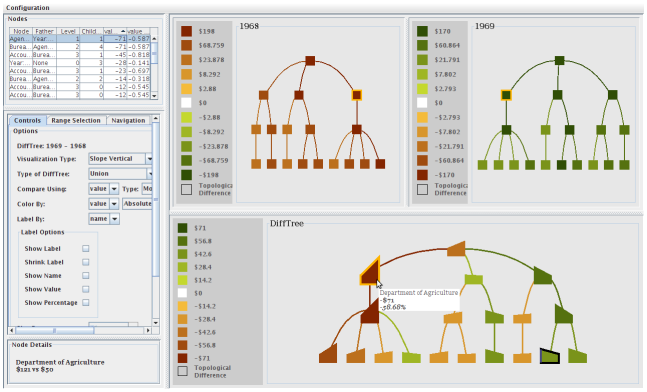

Figure 1: Example of the slope visualization representing a subset of a made up Federal Budget. The image shows the comparison of years 1968 vs 1969, where a cut of 58% was made on the "Department of Agriculture". Red nodes represent cuts, while green ones represent increases. The node with the black border represents a created node (topological difference).

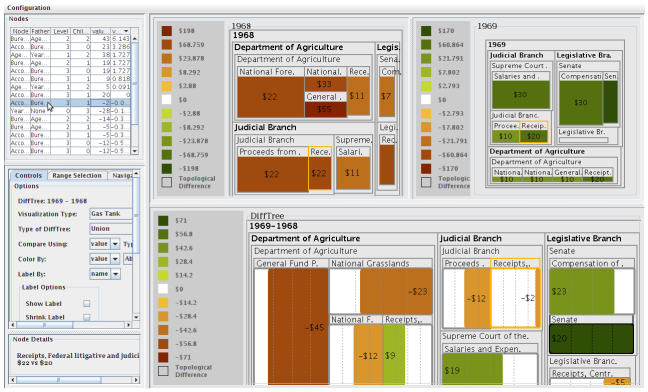

Figure 2: Gas Tank Visualization representing the data from Figure 1. It shows the absolute amount of change using color and a label, and the relative difference using the amount of space filled in each node. It also shows the topological differences using again a black border. This view also allows the selection of a node along all the compared trees and the diffTree to see its details, like it was done with the "Receipts, Federal litigative and judiciary" that decreased only on $2 from 1968 to 1969.

reduced (bigger on the compared tree on the left), while the increasing slope denotes and increased value (bigger on the compared tree on the right). The steepness of the slope represents the amount of change, and no slope means no change. The color represents the same amount of change using a gradient of tree user selectable colors.

For nodes that are only present in one of the compared nodes (topological differences) we used a black border as a special marker. An example of this visualization is shown on Figure 1.

Some of the slope visualization's strengths are that it shows differences in all the levels of the tree, that highlights topological differences changes and that displays the structure of the aggregated tree. However it works better with smaller trees, and can only represent one magnitude of change at a time (either absolute or relative). Because of this we developed a complementary visualization that we present in the next section.

3.2 The Gas Tank

The gas tank representation uses an space filling approach based on treemaps [3]. It represents changes in the leaf nodes of the diffTree, displaying at the same time the absolute and relative amounts of change. It is specially good to highlight the "biggest players" on the diffTree (nodes with the biggest values overall).

To create the gas tank we first take the two individual treemap nodes representations, we combine them and calculate the change, and finally we draw the difference as a filling portion of the area of the biggest of the original nodes. This way we avoid nodes of size zero. We then use this node representations of the leaves and aggregate them in the same way treemaps do to represent the hierarchy. Although the gas tank visualization only represent the changes on the leaf nodes, TreeVersity has a control that allows the selection of a level of the tree to represent, that way the gas tank diffTree is redrawn to represent only nodes in that level (or leaves on previous levels). An example of the gas tank representation with the same artificial budget data is shown in Figure2 on the right.

Like the slope visualization, the gas tank also has its strong and weak points. It is specially useful to compare the absolute and relative changes in the biggest nodes, but it isn't so helpful to represent the hierarchy of the tree, and it hides information like the amount of change in the interior nodes.

4 Acknowledgements

We thank the Fulbright International Science and Technology Scholarship for supporting John's doctoral studies. The Center for Integrated Transportation Systems Management (a Tier 1 Transportation Center at the University of Maryland) for partial support of this research. We also thank the Center for Advanced Transportation Technology Laboratory (CATT LAB), Michael Pack, Michael VanDaniker and Tom Jacobs for their suggestions and feedback.

References

[1] J. A. Guerra Gómez, K. Wongsuphasawat, T. D. Wang, M. L. Pack, and C. Plaisant, "Analyzing incident management event sequences with interactive visualization," in *Transportation Research Board 90th Annual Meeting Compendium of Papers*, 2011.

[2] K. Wongsuphasawat, J. A. Gomez, C. Plaisant, T. D. Wang, B. Shneiderman, and M. Taieb-Maimon, "LifeFlow: visualizing an overview of event sequences," in *Proceeding of the twenty-ninth annual SIGCHI conference on Human factors in computing systems. ACM*, 2011.

[3] B. Johnson and B. Shneiderman, "Tree-maps: A space-filling approach to the visualization of hierarchical information structures," in *Proceedings of the IEEE Conference on Visualization (Vis)*, pp. 284 – 291, IEEE, 1991.

[4] D. Fisher, R. Dhamija, and M. Hearst, "Animated exploration of dynamic graphs with radial layout," in *Proceedings of the IEEE Symposium on Information Visualization*, vol. 2001, pp. 43 – 50, IEEE, 2001.

[5] J. B. Kruskal and J. M. Landwehr, "Icicle plots: Better displays for hierarchical clustering," *The American Statistician*, vol. 37, no. 2, pp. 162 – 168, 1983.

[6] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou, "TreeJuxtaposer: scalable tree comparison using Focus+Context with guaranteed visibility," in *ACM SIGGRAPH 2003 Papers*, (San Diego, California), pp. 453–462, ACM, 2003.

[7] M. Graham and J. Kennedy, "Combining linking and focusing techniques for a multiple hierarchy visualisation," in *Information Visualisation, 2001. Proceedings. Fifth International Conference on*, p. 425–432, 2001.

[8] Y. Tu and H. Shen, "Visualizing changes of hierarchical data using treemaps," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 6, pp. 1286–1293, 2007.

[9] P. Bille, "A survey on tree edit distance and related problems," *Theoretical Computer Science*, vol. 337, pp. 217–239, June 2005.