Prioritizing Nodes in Hierarchical Visualizations with the Tree Cut Model

Rafael Veras* Christopher Collins†

University of Ontario Institute of Technology

ABSTRACT

We report on the development of a novel technique to select relevant nodes for presentation in hierarchical visualizations. It adapts the Tree Cut Model to the problem of determining the uneven deepest relevant level in a hierarchy, using it as a criterion for filtering data items. In a case study using Docuburst, we demonstrate how this technique can help creating better overviews of data by reducing visual clutter while highlighting relevant data.

1 INTRODUCTION

Hierarchical data, to a greater extent than its flat counterpart, naturally demands from visualization tools full support to the visual information seeking mantra: "overview first, zoom and filter, then details on the demand". Failing to support well these tasks can render interactive exploration infeasible. For large hierarchies, in particular, especial care should be taken in the design of the overview: while too much detail should be avoided due to clutter, if it is too high-level, little information can be grasped without drilling-down. A similar issue permeates the design of zooming, when each step might reveal too much or too little information, increasing the cost of interaction. So how to determine the proper increment for zooming? Moreover, are all branches of a tree equally relevant? These issues are only relevant when dealing with large, messy and imperfect hierarchies. Such data is abundant in the natural language processing domain, where hierarchy nodes and levels can be redundant or irrelevant (e.g., in hierarchical topic modelling).

In this work-in-progress report, we address the problem of determining adequate levels of detail in hierarchical visualizations with the ultimate goal of delivering more expressive views of large hierarchical data, informed by a data-centered measure of interestingness. Building upon a classic linguistic model originally developed for identifying selectional preferences of verbs [2], we identify uneven tree cuts that define appropriate levels of abstraction in diverse contexts. We adapt this technique to common hierarchical visualization tasks and demonstrate its value with a use case in the linguistic domain, applying it to the Docuburst visualization [1].

2 TREE CUT MODEL

The tree cut model is a generalization method based on the Minimum Description Length (MDL) principle, originally developed for generalizing values of a case frame slot for a verb [2]. Given a hierarchy of nouns and the frequency with which the leaves co-occur with a certain verb, Li and Abe's method estimates a horizontal uneven tree cut that represents an accurate and concise abstraction (in the MDL sense) over the set of nouns. In other words, each member of the tree cut is a class that best represents its subtree in a certain context (defined by the frequencies); for example, the class *bird*, whose children are *eagle*, *crow* and *hawk* might be a good abstraction when the node frequencies correspond to the co-occcurrence with the verb "to fly", while the higher level class *animal* might be more appropriate when considering the verb "to extinct".

That notion of appropriateness is quantified by the Description Length (DL), which incorporates the MDL principle of information theory: the best probability model is that which achieves the highest data compression. By acquiring a good estimation of the underlying probability distribution for the ocurrence of the signs, it is possible to find an efficient coding scheme. As such, the DL consists of the sum of the Parameter Description Lenght (PDL), which can be seen as an inverse measure of conciseness, and the Data Description Length (DDL), which can be seen as a measure of fitness to the data, or accuracy. Li and Abe proposed an efficient algorithm to find the tree cut that minimizes the description length, i.e., the one that provides the best balance between conciseness and accuracy.

We adapt the tree cut model to the more general case of determining the uneven deepest relevant level in a hierarchy, following the intuition that regularities in the values of nodes are inversely proportional to interestingness. The tree cut is used as a criterion for the visibility of subtrees, as all nodes under the tree cut are collapsed. Therefore, the model matches the visual exploration goal of finding exquisite behaviors in data by evidencing nodes with disparate frequencies. Below, we explain how to calculate the tree cut model of a hierarchical dataset.

Let S be a sample of observations of an event a for a set of data entities D (e.g., births per district), where every $d \in D$ has an associated score $f(d)$ and $|S|$ is the total number of observations (e.g., total number of births). Each data entity is the leaf of a tree where the inner nodes represent higher-level classes (e.g., city, province, region). A tree cut model M is described as

$$M = (\Gamma, \theta) \tag{1}$$

where Γ is a set $[C_1, C_2, ..., C_{k+1}]$ of internal/leaf nodes that dominate all of the leaf nodes exhaustively and disjointly, and θ is a probability vector of the same length such that $sum(\theta) = 1$. The probabilities of the nodes under a tree cut are smoothed, i.e., the probability of a tree cut member is distributed uniformly across all nodes dominated by it.

Recall that the description length is given by:

$$DL = PDL + DDL \tag{2}$$

The parameters of a tree cut are estimated using Maximum Likelihood Estimation (MLE). Thus, DDL is calculated by:

$$DDL = -\sum_{d \in D} log(P_{\hat{M}}(d \mid a) \times f(d)) \tag{3}$$

$P_{\hat{M}}(d \mid a)$ is the probability of an entity given the event (or attribute) in consideration, and can be interpreted as the probability of d as estimated by the tree cut. It is obtained by normalizing the probability of the class C dominating d in the tree cut:

$$\hat{P}(d) = \frac{1}{|C|} \times \hat{P}(C) \tag{4}$$

where $|C|$ denotes the number of entities (leaves) under C. For each $C \in \Gamma$:

$$\hat{P}(C) = \frac{f(C)}{|S|} \tag{5}$$

*e-mail: rafael.verasguimaraes@uoit.ca

†e-mail: christopher.collins@uoit.ca

where $f(C)$ represents the total score of instances in class C. Finally, the PDL is given by:

$$PDL = \frac{k}{2} \times log|S| \tag{6}$$

We refer the reader to Li and Abe [2] for an algorithm to find the tree cut that minimizes the description length in a tree.

The original tree cut model is good for extracting a static summary of the target tree, but is not particularly useful in an interactive application, where users navigate the tree at different levels of granularity. To introduce more flexibility, we follow Wagner [3] and introduce a free parameter into the DL calculation that influences the specificity of the tree cut:

$$DL = PDL + W\left(\frac{log|S|}{|S|}\right) DDL \quad (C > 0) \tag{7}$$

where $log|S|/|S|$ corrects a disparity between the complexities of PDL and DDL, which caused the latter to grow faster as the input size increased, and W is a free weighting factor, directly proportional to the specificity of the cut (by decreasing the importance of the PDL) (Figure 2).

3 Case Study

We apply the tree cut model to Docuburst, a tool that allows the exploration of documents through a hierarchical visualization of the semantic categories extracted from them. Docuburst displays the WordNet hierarchy of nouns, containing over 80,000 nodes, using size and brightness to encode the frequency of the categories in the target document. Figure 1, top, shows the highest level overview (rooted at *entity*, the topmost category) of the 2008 US presidential debate transcript, which featured John McCain and Barack Obama among other candidates. In this view, a filter is set to display only the first 13 levels of the hierarchy. Note that the labels of many nodes are not drawn due to space constraints. The view seems very cluttered and the density of nodes affects the performance. Displaying only the first levels would be a poor option, as the higher level categories (e.g., abstraction, object) do not convey sufficient information about the subject matter. In Figure 1, bottom, we demonstrate the result of applying a tree cut filter on the same view. The filter preserves the visibility of only the nodes that lie above or along the tree cut. Note the view is much less cluttered, and that reduces the rendering overload. In particular, attention is drawn to the nodes that have the most distinguishing power. Figure 2 portrays the progressive behaviour of zooming, step-by-step, as represented by the different node colors. Instead of revealing a new even hierarchy level at each step, we increase the specificity of the tree cut by incrementing the W parameter in function of the "zoom in" action, causing entire subtrees to be revealed asymmetrically.

Figure 1: Comparison between the traditional docuburst view (top) and the filtered view using the tree cut model (bottom), showing the semantics of the 2008 presidential debate. In both versions the 13th hierarchical level is the deepest.

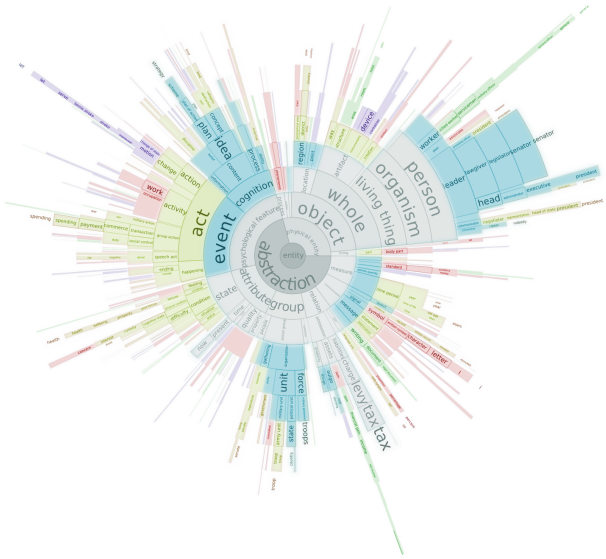

Figure 2: Levels of zoom detail color coded. The more external, the higher the value of the W parameter.

4 Conclusion

We conclude that the Tree Cut Model has great potential to solve some of the problems that affect the visualization of large hierarchies, such as rendering meaningful overviews and realizing effective zooming. As future work, we plan to apply this technique to other types of visualizations, such as treemaps, and evaluate how the selected nodes match people's intuition in terms of relevance.

References

[1] C. Collins, S. Carpendale, and G. Penn. Docuburst: Visualizing document content using language structure. In *Computer Graphics Forum*, volume 28, pages 1039–1046. Wiley Online Library, 2009.

[2] H. Li and N. Abe. Generalizing case frames using a thesaurus and the MDL principle. *Computational Linguistics*, 24(2):217–244, June 1998.

[3] A. Wagner. Enriching a lexical semantic net with selectional preferences by means of statistical corpus analysis. In *ECAI Workshop on Ontology Learning*, 2000.