Interactive Learning for Hierarchies of Concepts

Adwin Jahn[1], Ryan Maas[2], & Harley Montgomery[2]
[1]Department of Electrical Engineering
[2]Department of Computer Science & Engineering
University of Washington
{maas,wmonty}@cs.washington.edu,
adwin596@gmail.com

ABSTRACT
We present a visualization system for an algorithm which learns hierarchies of concepts by crowdsourcing answers to binary questions about the relationships between objects. A hierarchy is represented as a tree, where a node is considered a subtype of its ancestors. The algorithm learns a distribution over all possible hierarchies, but only returns the *maximum a posteriori* (MAP) hierarchy, which may not be representative of the entire distribution. Previous visualizations also do a poor job of showing how relationships change in response to crowdsourced answers. Our tool tackles the first issue by prominently encoding the uncertainty of the edges in the tree and by allowing to user to hover over a node to see other possible parent nodes, a process we call "edge highlighting". The second issue is addressed with an interactive session that walks the user through the process of the algorithm, highlighting the changes between relationships that occur after a question has been answered.

1. INTRODUCTION
Hierarchies of concepts have applications in many fields of computer science, from recommendation systems to natural language processing [2][7]. However, building trees to represent these hierarchies requires substantial human input, which can be a time consuming process. While fully automated processes have been proposed, the accuracy of built trees improves substantially with even small amounts of human direction, leading researchers to explore crowdsourcing solutions.

Using services like Amazon's Mechanical Turk, non-experts can provide input that aids the building of trees with automated algorithms. We consider one such approach which models the probability distributions of hierarchies based on responses from Mechanical Turk, providing an estimate of the most likely hierarchical tree for a given domain of concepts [9]. In particular this algorithm only requires binary responses to whether one concept is among any descendants of another concept, rather asking questions about explicit `parent->child` relationships. During training, the algorithm maintains a posterior distribution over hierarchies, as well as the probability of individual `parent->child` edges.

While this is an effective algorithm, the accuracy of the estimated trees depends on the number of iterations of questions asked. Furthermore, noise or conflicting information in the responses of non-expert Mechanical Turk workers can manifest itself in trees that are un-intuitive or contain errors according to experts. This is most evident when visualizing the most likely trees produced by the algorithm, with one example shown in Figure 1. The above example shows a tree hierarchy of body parts built with crowd-sourced data. Most of the relationships are accurate, but there are several concepts which are incorrectly assigned, like "ring finger" as a direct descendant of "upper body" but not of "hand." The above visualization con-

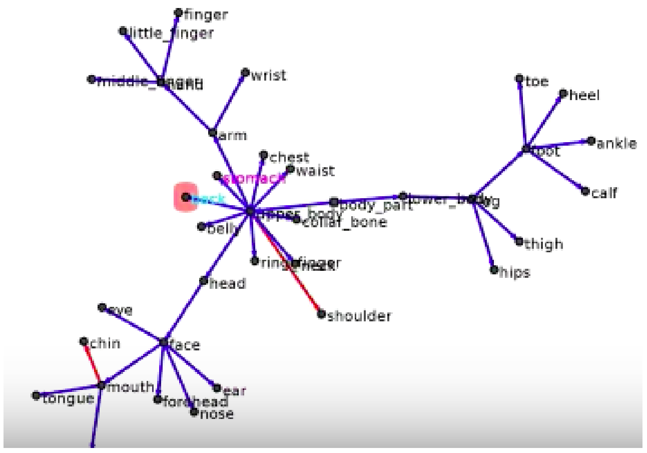


Figure 1: Visualization of a tree hierarchy of body parts made using the algorithm in [9]. The diagram shows a hierarchy of connected body parts, but on inspection has several error nodes such as "ring finger."

veys no information about the model's confidence in the incorrect nodes, or which questions might be asked to improve the most likely tree. Thus an expert reviewing the tree by this visualization lacks information that's present in the underlying algorithm.

We approach the challenges of visualizing the trees produced by this algorithm, and capturing the inherent probabilistic uncertainty of the method. Such a visualization would help experts evaluate the performance of the algorithm any a particular iteration, and could potentially provide helpful information for crowd-sourcing workers to provide better responses.

2. RELATED WORK

There has been extensive work on visualizing network hierarchies in the literature. Early work focused on which types of visualization are best suited for different types of networks. [4], [10], and [6] survey a variety of tree visualizations in practice, including how various visual encodings can influence the effectiveness of the tree. [3] proposes more readable tree maps, adding more compact rectangles and frames to emphasize existing hierarchy.

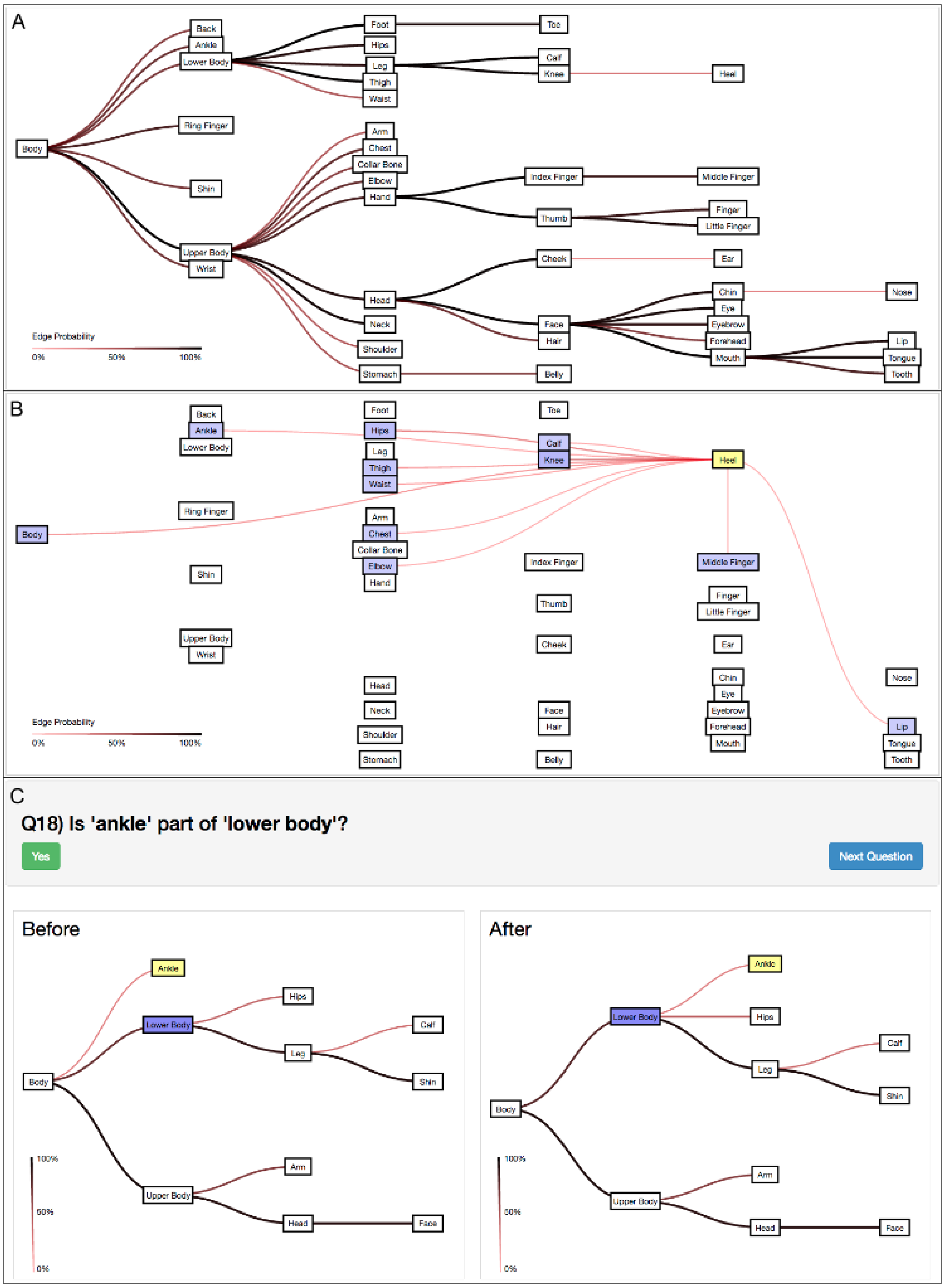
Particularly relevant to us are research on dynamic visualizations of trees and methods of visualizing uncertainty. [5] considers methods for displaying changes in trees, particularly for nodes which have uncertainty relationships with other potential parents. [1] explores dynamic network visualizations and animations, termed network movies. It makes the distinction between static flip books, where nodes remain in place while edges move, and movies, where nodes move as a function of changes in relations. [8] examines how structural uncertainty can be conveyed in tree diagrams, which is one of the most relevant papers to our works visualization of probabilistic edge visualization.

3. METHODS & RESULTS

We addressed the issue of uncertainty in two major ways, one static and one dynamic (Figures 2A-B). For the static visualization, we encode the probability of the direct `parent->child` relationship in the color, thickness, and opacity of the edge between `parent` and `child`. Edges with lower probability appear redder, thinner, and more transparent, in order to give the appearance that they are "unstable". We chose to triply encode the edge strength to emphasize the "instability" of weaker edges, as we personally found that any single encoding did not give the desired effect.

For the dynamic visualization, we use a method that we denote "edge highlighting". When the user hovers over a node all current edges disappear, and new edges appear showing all `parent->child` edges above a given threshold, where the node acts as the `child`. Edges shown during edge highlighting are encoded with the same attributes as the static visualization, and the `parent` nodes are also shaded according to the edge probability. This allows the user to quickly see which other `parent` nodes are reasonable, and how they compare to the relationship in the current hierarchy.

We also built an interactive application which allows users to see how the algorithm progresses as they answer the same questions that would



Q18) Is 'ankle' part of 'lower body'?

Figure 2: Screenshots of Visualization

A) An example of a hierarchy visualized with our tool after training on crowdsourced answers. Uncertain edges, such as `knee->shin` can be immediately detected. B) “Edge highlighting” demonstration, showing the other `parent->child` edges which are above the threshold. C) The interactive session shows the user how their answer affected the hierarchy.

be given to crowdsourcers (Fig. 2C).[1] The user is presented with the current tree, visualized as described above, with the `parent` and `child` nodes highlighted. When the question is answered, the tree transitions to the next iteration smoothly, and then displays the tree before and after the question was answered. This allows the user to track how their answer affected the hierarchy, with a focus on the nodes in question.

4. DISCUSSION

As we continue to turn to probabilistic models for data analysis, we must address the issue of visualizing uncertainty effectively, since the MAP solution may not represent the entire distribution (or because approximate inference might return a suboptimal solution). Our system was a first attempt at piercing through the "fog of uncertainty" by allowing the user to probe the MAP solution and see relationships which would not be otherwise apparent. Therefore we believe that both the edge encoding and edge highlighting are important aspects of the visualization. The edge encoding makes it immediately apparent which relationships are relatively uncertain, and therefore which nodes are worth inspecting with the edge highlighting tool.

This visualization system can play a role for both the experienced user as well as the new user. For the experienced user, this could serve as a debugging tool for examining a crowdsourced hierarchy and understanding where/why uncertainty still exists. For example, in Fig 2 it is clear that the edge from `knee` to `shin` is uncertain. Probing further with edge highlighting, we see that `shin` has a relatively weak connection to many nodes, and therefore more answers are needed before the node can be placed with certainty.

[1]Due to time constraints, our collaborator was unable to get us an interactive executable of the algorithm. To sidestep this issue, we use a previous run of the algorithm, and assume that our user will give the same answer as the Mechanical Turk user. This means that our application is not *really* interactive, but this simplification suffices for a demonstration.

For the new user, this tool can be used to understand why the MAP hierarchy defies their expectations. For example, during the early stages of training the tree is relatively uncertain, and many nodes move around in response to a single question. Given just the MAP tree, it might be unclear to the new user how a single question can alter the tree so dramatically. However, our visualization makes it apparent that *all* of the connections are uncertain in the beginning, so unexpected changes can be explained by the noisy approximate inference (and ignored until the hierarchy has become more certain). Furthermore, by highlighting the nodes from the question in both the "before" and "after" images, it is more clear exactly how a single question affects the relevant nodes.

5. FUTURE WORK

Our visualization system provides the user with a tool to probe the uncertainty in the distribution over possible hierarchies. However, this was just a first attempt at such a system, and in future work we would continue to add features with this goal in mind. In particular, our tool focuses on examining local relationships (i.e. `parent->child`), but it would be interesting to allow for a more global examination of the distribution. For example, given more access to the algorithm internals, we could generate new (approximate) MAP trees on the fly, or sample several trees based on their likelihoods. For large hierarchies this might prove unreasonable, which raises the question of how to effectively explore uncertainty on a global and local level simultaneously.

Another future direction would be to examine how these visualizations can be used in conjunction with the crowdsourcing algorithm to improve the quality of the learned hierarchies. Again, this could be addressed at both the level of the experienced user or the crowdsourcer. An experienced user could use this tool to track the current progress of the algorithm, and potentially correct mistakes online or tell the algorithm which nodes should be focused on in future questions. For the crowdsourcer, an interactive tool could allow them to see how their answers affect the current process, which could increase engagement and improve the quality of answers. Of course, it's also possible that it would only confuse the crowdsourcer, so a future study would provide valuable information for future iterations of the system.

6. REFERENCES

[1] S. Bender-deMoll and D. McFarland. The art and science of dynamic network visualization. *Journal of Social Structure*, 7(2):1–38, 2006.

[2] S. Bloehdorn, A. Hotho, and S. Staab. An ontology-based framework for text mining. In *LDV Forum âĂŞ GLDV Journal for computational linguistics and language technology, 2005, Vol.20, No.1*, pages 87–112.

[3] M. Bruls, K. Huizing, and J. V. Wijk. *Squarified treemaps*. Springer, 2000.

[4] M. Graham and J. Kennedy. A survey of multiple tree visualisation. *Information Visualization*, 9(4):235–252, 2010.

[5] J. Guerra-Gomez, M. Pack, C. Plaisant, and B. Shneiderman. Visualizing change over time using dynamic hierarchies: Treeversity2 and the stemview. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2566–2575, 2013.

[6] I. Herman, G. Melançon, and M. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000.

[7] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset.

[8] B. Lee, G. Robertson, M. Czerwinski, and C. Parr. Candidtree: visualizing structural uncertainty in similar hierarchies. *Information Visualization*, 6(3):233–246, 2007.

[9] Y. Sun, A. Singla, D. Fox, and A. Krause. Building hierarchies of concepts via crowdsourcing. *arXiv preprint arXiv:1504.07302*, 2015.

[10] T. T. Barlow and P. Neville. A comparison of 2-d visualizations of hierarchies. In *infovis*, page 131. IEEE, 2001.