Visualization of Semantic Networks to Understand Alzheimers Disease

Bryce Johnson (bcjohnson7.wisc.edu)
Department of Computer Science, 1210 W Dayton St,
Madison, WI 53706

Abstract

Semantic networks have been shown to offer insights into the early detection and understanding of the neurological processes of Alzheimer's Disease (AD). However, interpretable visual analysis of these networks has previously been intractable due to the inability to visualize complex networks on a 2D plane. In this work I consider three possible ways (traditional methods, UMAP, and topological analysis) to alleviate this bottleneck. Of the three, topological analysis performs the best as a low fidelity visualization for quick interpretation of large networks. Since the topological analysis is highly dependent on a continuous node "fitness", three possible fitness functions: fluency list order, word frequency, and a combination of the two are studied. Fluency list order fitness is determined as the best representation for semantic networks since it does the best at displaying clusters. Differences in the topology of AD and NC network structure are also discussed and an insight into the neurological processes that causes AD is proposed.

Keywords: visualization; semantic networks; Alzheimer's disease; Partáy Algorithm

Introduction

It has long been debated whether the degradation of neurological processes that cause Alzheimer's disease (AD) is due to the degradation of the semantic memory store or the failure of the information retrieval mechanisms from a working semantic store. Interpretable graphical models offer insights into this debate.

Previous work has aimed at generating these networks from fluency lists (Zemla & Austerweil, n.d.). Fluency lists are usually used to quickly identify probable AD patients. One example of a list would be to have a patient name as many animals as they can in one minute. AD patients will usually name less items and repeat more items than normative control (NC) patients. Interestingly, the words that patients list are usually not independently distributed. Instead, words are generated in a systematic order based off correlations to the previous words listed. If correlations are shown as edges and words as nodes the result is a graphical model known a semantic network. (Fig. 1).

In the work of Zemla and Austerweil (n.d.) it was found that graph properties such as node density, node degree, and small worldness, can be useful for prediction of AD, however, the properties alone are not as interpretable as reading the graph itself. In their work this was not possible due to many networks having complicated structure, making visual analysis intractable. In this work we consider three possible ways to alleviate this bottleneck and produce stronger visualizations. The first uses classical techniques such as the spring and shell layout. The second, UMAP, attempts to show linear separability between AD and NC networks using network structure and then network properties as features. The final technique aims at visualizing the clusters of the graph using a topological analysis by assigning a continuous fitness to each of the nodes in the network. This work explores four possible fitness functions, specifically list, word, and combination of word and list specific fitness. List specific fitness is determined as the best representation for semantic networks since it does the best at displaying clusters. With these results comparisons of AD and NC network structure could be done using visual classification and possibly lead to insights in the neurological processes that cause AD.

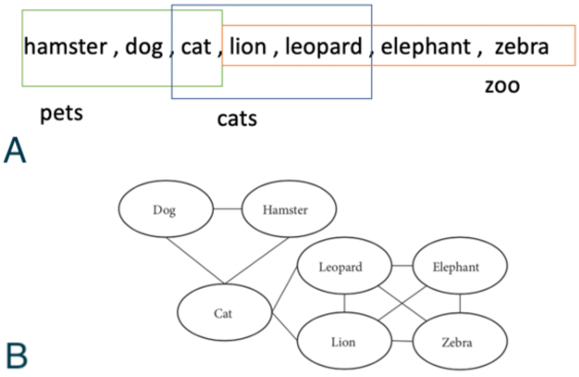

Figure 1: (A) Example semantic fluency list. Clusters: green–pets, blue–cats, orange–zoo. (B) Ideal semantic network. ((B) taken from Zemla and Austerweil (n.d.))

Methods

Semantic fluency lists were collected on a year by year basis for 158 participants at the UCSD Alzhiemers Disese Research Center. Every year, each participant was given a diagnosis of either AD or NC based on the National Institute of Neurological and Communicative Disorders and Stroke-Alzheimer's Disease and Related Disorders Association (NINCDS-ADRDA) scale. Associated networks were made for a patient by (Zemla & Austerweil, n.d.) using the unsupervised U-INVITE algorithm using all available semantic fluency lists for a specific diagnosis. The U-INVITE approach is inherently Bayesian, as it assumes a prior on the graphical structure and a few hyperparameters that are used in the U-INVITE algorithm. For the graphical prior they choose an unweighted and undirected semantic network constructed from free associated norms compiled by University of South Florida.

Visualization of the these semantic networks was attempted using three types of algorithms (1) node layout algorithms which changed the positions of nodes for better representation (2) UMAP - contionous dimensionality reduction technique, (3) a topological analysis of the graph. As Zemla and Austerweil (n.d.) found, networks with higher numbers of nodes tend to have more lists, but have the same properties as smaller networks (which come from fewer semantic fluency lists) for similarly diagnosed patients. As such it would be expected that AD networks would have similar visualized topology to another AD network with less fluency lists.

Layout Algorithms

All layout algorithms were provided by NetworkX. The three criteria used to judge the effectiveness of a visualization were ability to (1) read node values, (2) interpret edges between nodes, (3) identify possible clustering of nodes.

The spring layout calculates the positions of the nodes as if edges are springs (pulling nodes together) and nodes are charge particles (repulsing each other apart). The ideal positions of the nodes is found by equilibrating the system by moving the node positions in a 2D plane. (Gibson, Faith, & Vickers, 2013)

The second layout algorithm was the circle layout, which placed all nodes into a circle. Analysis of both of these layouts was done for simple and complex AD and NC networks.

UMAP

UMAP is a dimensionality reduction technique that aims to find a 2D manifold which all points live and project that manifold onto the 2D plane. (McInnes, Healy, & Melville, 2018) UMAP only works with many data points and graphical structures do not provide this type of representation easily. In order to use UMAP all possible patients' networks must be considered. Then ideally, in the 2D plane they will show linear separability between the AD and NC networks. A single data point was made using two representations: a graph structure representation and a graph properties representation.

The graph structure representation was made by enumerating an adjacency matrix to include node values from all possible networks (a global adjacency matrix). The graph properties representation was found by using the 20 properties as calculated in Zemla and Austerweil (n.d.) for a single graph. UMAP was done on both of these representations and shown on a 2D plot. Large hyper parameter sweeps were performed for the minimum number of neighbors and the minimum allowed distance to get the best visualization for the UMAP algorithm.

Partáy Algorithm to Visualize Topology

Unlike node placement algorithms the Partáy algorithm does not suffer from the constraint that networks become more clustered visually with increasing number of nodes. Instead the algorithm maps nodes into clusters in the 2D plane and shows them as ”peaks” (similar to how one would look at peaks of a continuous function) described by a continuous fitness metric. Originally the algorithm was made for exploration of configurational landscapes of molecules. (Pártay, Bartók, & Csányi, 2010) Although we lose knowledge about the collapse in space by not using a Bayesian sampling routine, we could use this algorithm to display complex systems like semantic networks.

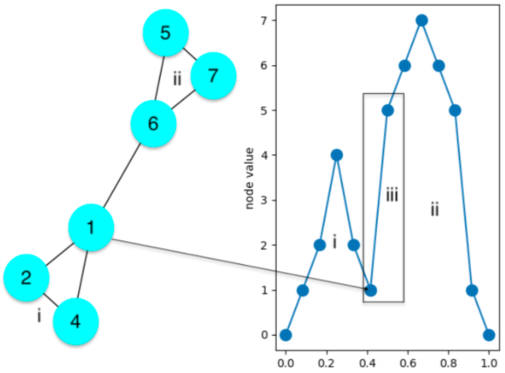

Figure 2: Toy network (left) with a the corresponding visualization using the Partáy algorithm (right). Node fitness corresponds to the nodes number.

The algorithm begins by defining a fitness for each of the nodes in the graph. In this work many fitness functions were considered, including list, word, and the combination of list and word specific fitness. The algorithm continues by pruning the nodes of the network with the lowest fitness first. A toy model of this is shown in Fig. 2. Node 1 is pruned first, resulting in two remaining clusters, (ii) and (i). Clusters (i) and (ii) making up 0.4 and 0.6 of the remaining nodes, respectively. This results in a split shown by (iii) where each peak has width proportional to the number of nodes in that cluster. The algorithm continues pruning nodes recursively, until their are no remaining edges, at which point the node with the highest fitness value is chosen to be the peak. For the purposes of interpretability when working with semantic networks, any node that is chosen as a peak, the corresponding word will by shown on top of its peak. Although information is lost from not knowing what nodes were removed nearby, inspection of the peak and topology can reveal roughly what cluster of words is in that corresponding peak.

A summary of the fitness functions and how they computed fitness is shown in Table 1. The motivation for the list specific fitness functions was that the ordering of words in the list should be highly correlated to clusters in the words.

Word frequency was assumed to be a prior and frequency values were not normalized which was taken to be an error. The word frequency fitness function aims at finding peaks of extremely common occurring words. However, as it will soon be shown, words commonly don't form separate word by word frequencies.

Under the Bayesian perspective the list specific fitness can be interpreted as the likelihood, and the word frequency can be interpreted as the prior for a given word. To account for the error to normalize the prior, a decreasing exponential was applied to the list increasing fitness to give more variability in the magnitude of the likelihood. The posterior probability distribution wasn't normalized, however dividing by a constant wouldn't change the results so this was ignored. The assumption was that words that have a higher prior are more likely to be interconnected, where the likelihood function is more likely to group clusters of specific words spoken near one another.

If a word appeared in a fluency list multiple times, only the fitness of the first value found chronologically was taken. If a word was seen in multiple fluency lists for a specific patient the fitness values were accumulated.

Word values from the the raw semantic fluency list data did not always directly match those found in the pregenerated AD and NC networks (e.g. lions instead of lion, drom dry instead of dromedary). The words in the semantic fluency list were replaced by the node with the minimum Levenshtein edit-distance for each word. This was noted as a possible error and upon initial visual inspection no values changed incorrectly for the few semantic networks studied in this paper.

Table 1: Node specific fitness description, the equation, and a fluency list example (Lion, Cat, Dog).

description	equation	(Lion, Cat, Dog)
list increasing	$P(L\|W)_i$	(0,1,2)
list decreasing	$P(L\|W)_d$	(2,1,0)
word frequency	$P(W)$	$(10^{-5}, 10^{-5}, 10^{-4})$
word+list	$P(W)*$ $\exp(-P(L\|W)_i)$	$(0, 10^{-5}, 2*10^{-4})$

Results and Discussion

Layout algorithms

Visualizations of one AD semantic network with the spring layout is shown in Fig. 3. Due its small structure this semantic networks met the first three criteria (**Methods**) of an effective visualization. Many AD networks had a highly connected or cyclic structure in the center with many long chains connected to the center cluster. The creation of many chains could be due to the fact that the graphs were generated from multiple semantic fluency lists, where each chain is a fluency list from a seperate year.

When working with large AD networks ($>$ 20 nodes) the visualizations become cluttered on a 2D plane (as shown in Fig. 4). In order to interpret these networks larger canvas sizes, smaller node values, or interactive resources (e.g. Cytoscape) are necessary. However, these approaches require much more user analysis than a simple glance at the page, since it is necessary to trace out all the nodes and their possible edges. When looking at many networks this leads to a slow development cycle, therefore a more robust algorithm that produces low fidelity visualization for quick analysis would be useful.

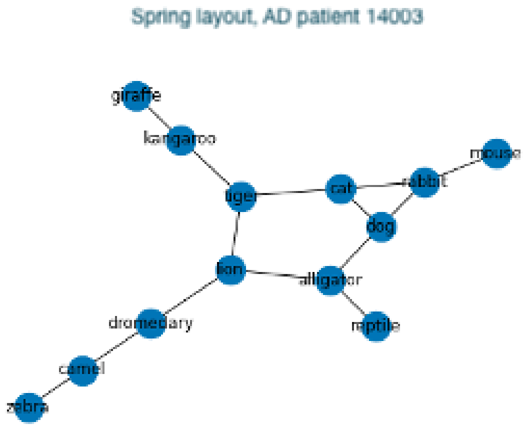

Figure 3: Spring layout visualization for patient 14003 who has AD.

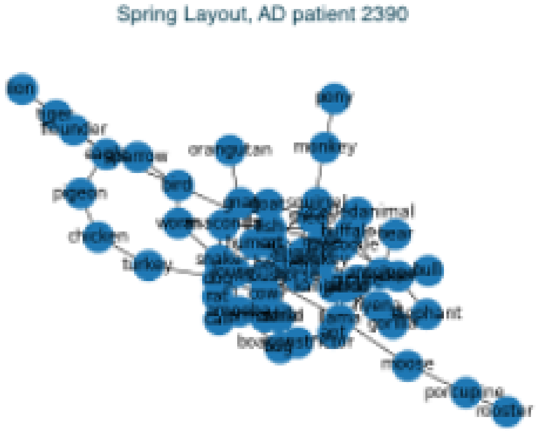

Figure 4: Complex network using spring layout for patient 2390 with AD.

As expected NC graphs visually had more nodes than AD graphs (although not always). They usually contained a more compact and connected center clusters compared to AD networks.

The second layout tried, a circle layout, worked better for large systems on small canvas sizes since all nodes were visible. However, as shown in Fig. 5 much of the network topology is lost. For the circle layout, NC networks would contain more distinct edges that crossed the entire circle than AD edges, and usually contain more nodes around the perimeter, as expected from analysis of network properties.

Many other algorithms were tried, including the spectral, random, and kamanda-kawai layout. These layouts either provided no interpretable information or were repetitive of either the spring or shell layout.

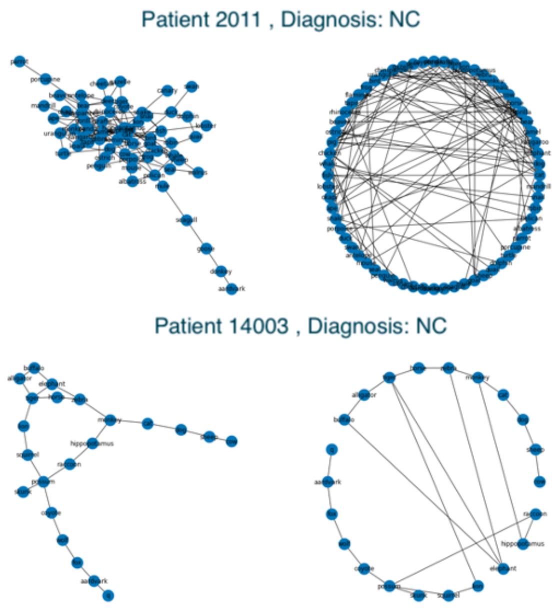

Figure 5: Spring (left) and circle(left) layout for NC networks patient 14003 (top) and 2011 (bottom).

UMAP

Results of the UMAP for the graph structure and graph property representation are shown in Fig. 6. The graph structure result considered was not successful because the input features was to sparse. With 605 unique nodes, the adjancency matrix had 366025 features, far to many to allow for easy convergence using UMAP. Since the graphical property was also not linearly separable when projected in UMAP, but seperable when done with linear regression, this suggests strongly that AD and NC graphs do not exist on a linearly seperable 2D manifold.

Partáy algorithm to Visualize Topology

Results for the Partáy algorithm for the AD graph in Fig. 3 are shown in Fig. 7 for word specific fitness values. It is hypothesized that the ideal topology of this AD graph will have a few competing maxima, mainly corresponding to a few of the outer node chains. However, word specific fitness favors interconnected nodes too much, so it prunes from the outsides resulting in a single large peak at the center that favors common words like dog and cat.

The second fitness tried is the list likelihood increasing node fitness (shown in Fig.8). This showed multiple competing maxima, quite in line with the expected structure of the graph. We see one large cluster for rabbit in the center corresponding to the large cycle. Other maxima such as lion and reptile correspond to the chains connected to the cyclic center. Its not a perfect match, some clusters are guaranteed to be lost, but it shows the topology of the graph on a lower dimensional plane.

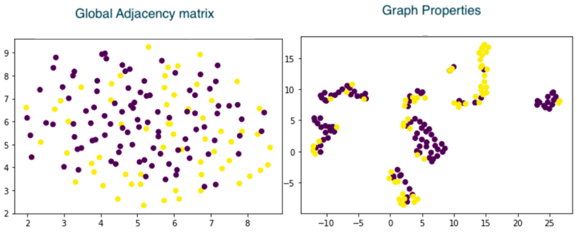

Figure 6: UMAP embedding for graph structure representation(neighbors=20) and graph property representation (neighbors=5). Yellow dots represent AD patients while purple dots represent NC patients. Both embeddings used the euclidean distance metric with a minimum distance of one.

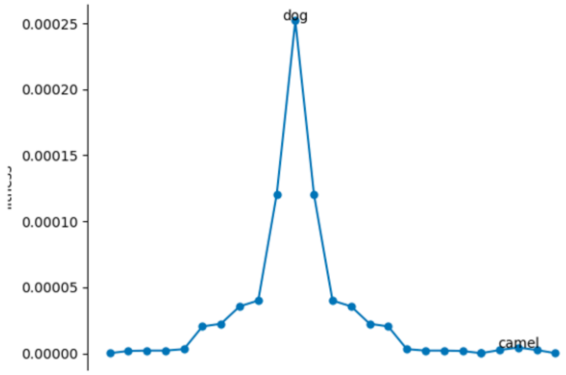

Figure 7: Partáy algorithm for word frequency fitness on patient 14003.

If we analyze the fitness values of specific nodes (Fig. 9) we see that the nodes near the extremeties have the lowest fitness, and they increase in value as they go towards the center. This type of removal pattern could suggest that AD patients have a warm-up period of random correlations, followed by a period of listing words which are related to each other similar to an NC patient. This claim is purely skeptical, however, if true, it could be possible evidence of information retrieval failure. The semantic store still exists to make correlations, it just takes a bit of a warm up period to get there. However, this phenomena could also be explained by the the combination of many semantic fluency lists to get fitness values. The different node chains could correspond to different lists and it is just coincidence that the nodes in the center cluster occur later in the list.

The decreasing fitness value (Fig. 10) shows the exact same information in its plot as list increasing except in reverse order. The inner nodes are removed first resulting in extremity chains being taken as possible clusters. That is why we see one large volume followed by many small peaks on top of it. The information conveyed in this plot is the same as the increasing fitness simply with a different representation.

The Bayesian fitness function (Fig. 11)shows a competi-

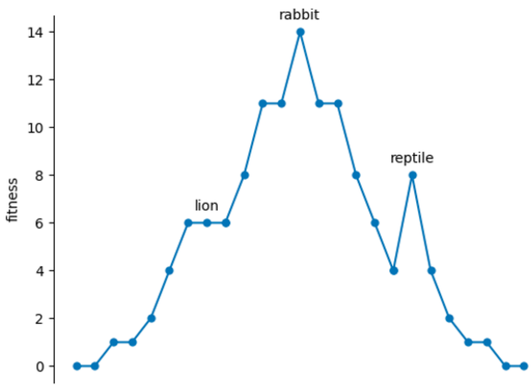

Figure 8: Partáy algorithm for list increasing fitness on patient 14003 with AD.

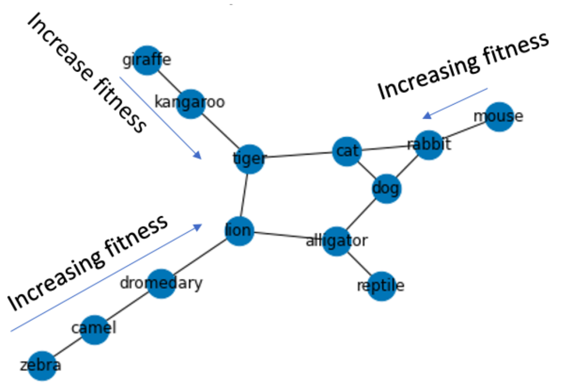

Figure 9: Visual representation of node fitness values for list increasing fitness of patient 14003 with AD.

tion between interconnectedness (prior) and extremity values (likelihood). These clusters weren't perfect in determining the top node values, in that they had some nearby nodes as cluster peaks (e.g. tiger and giraffe), overall it still provides a strong representation of the results. However, due to the complexity of the function and error to normalize the prior it is hard to determine the behavior of the function as it relates to the topology of the graph.

Overall, the prior results suggest that humans don't generate semantic fluency lists with the most common word and then deviate from that with related least common words. Instead, humans form clusters of words more relevant to the likelihood function, where words are grouped based off their ordering in the fluency list, indicating the increasing list fitness the best function of the four explored here.

In this study we only compared to a single NC graph of a different patient. Future work could expand on this by comparing to many NC and AD graphs to see if their is a pattern in the topological structure. The results of the Partáy algorithm on a single NC network and the corresponding spring

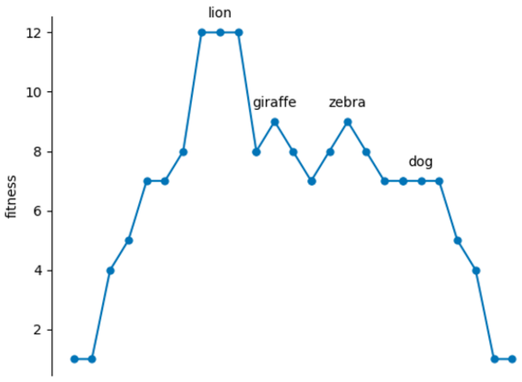

Figure 10: Partáy algorithm for list decreasing fitness on patient 14003 with AD.

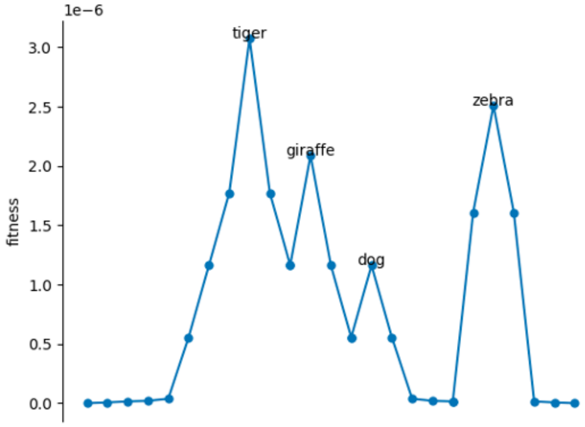

Figure 11: Partáy algorithm for combined list and word fitness on patient 14003 with AD.

visualization are shown in Fig. 12 and Fig. 13, respectively. These figures show that highly connected nodes results in a large contraction. However, there still exists many disjoint clusters, most notably the ”guppy” cluster (highlighted in red in both figures.) This graph topology looks noticeably different from the AD graph, in that its a rapidly contracting fitness, where as the AD graph has a large volume followed by a few peaks on top. More comparisons of NC and AD networks are necessary to reach stronger conclusions.

Limitations and Takeaways The largest limitation of using the Partáy algorithm is the need for a novel and useful fitness function which correlates well to the clusters that want to be visualized. If the fitness was just randomly assigned (as could unknowingly be the case for some fitness functions), the visualized topology would be very uninformative. In contrast if done correctly as shown here for a few networks it possibly fills a necessity in the graph visualization community to show network topology on a 2D plot which is usually

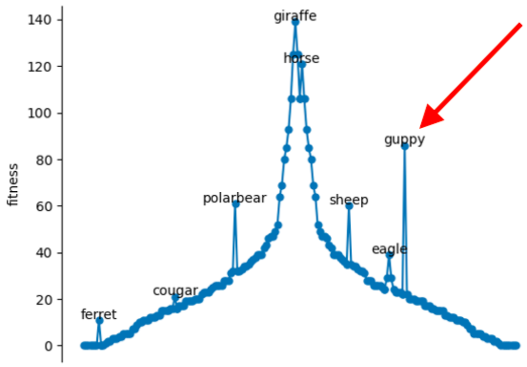

Figure 12: Partáy algorithm for list increasing fitness on NC patient 2004. The guppy cluster is pointed out by a red arrow.

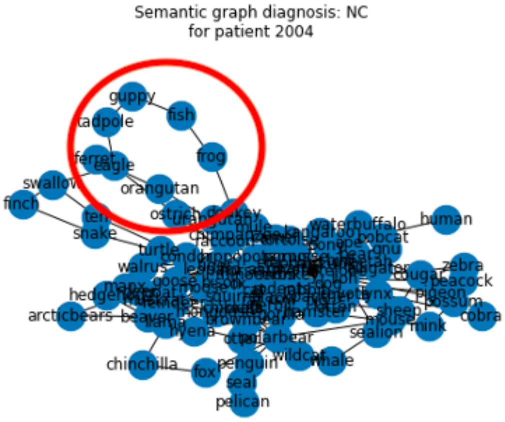

Figure 13: Spring layout for NC patient 2004. The guppy cluster is circled in red.

only done sufficiently with interactive plots.

Another constraint of the algorithm is analyzing its effectiveness. Perhaps, by the time one analyzes exactly what the output of the fitness function is doing, the same time could've been spent analyzing specific edges in Cytoscape.

The final limitation is that this also hasn't been tested on large networks with thousands or millions of nodes. This would be necessary if this type of approach were to translate into other graphical communities. If their were to many competing peaks, then the 2D plot would become cluttered, no different than a spring layout.

The overall take away is that the the visualization can imply network structure, but it is highly dependent on the fitness function of choice.

Reproducibility

All code for the project can be found at `https://github.com/brycejoh16/cs841`. Direct all questions to **bcjohnson7@wisc.edu**.

Conclusion

In this work I demonstrate many techniques to represent complex networks on a 2D plane. These techniques include layout algorithms, UMAP, and the Partáy algorithm. I show the Partáy algorithm is superior to the other two approaches when aiming to produce low fidelity visualizations with a interpretable fitness function. Four fitness functions were studied, and I show that list increasing fitness has superiority in terms of interpretability and correctness in displaying the topology of a single AD graph. The comparison of a single AD graph is done with an NC graph to see the differences in topological structure. Possible processes governing the neurological degradation of AD were also suggested.

References

Gibson, H., Faith, J., & Vickers, P. (2013). A survey of two-dimensional graph layout techniques for information visualisation. *Information Visualization*, *12*(3-4), 324-357. Retrieved from `https://doi.org/10.1177/1473871612455749` doi: 10.1177/1473871612455749

McInnes, L., Healy, J., & Melville, J. (2018). *Umap: Uniform manifold approximation and projection for dimension reduction.* arXiv. Retrieved from `https://arxiv.org/abs/1802.03426` doi: 10.48550/ARXIV.1802.03426

Pártay, L. B., Bartók, A. P., & Csányi, G. (2010). Efficient sampling of atomic configurational spaces. *The Journal of Physical Chemistry B*, *114*(32), 10502-10512. Retrieved from `https://doi.org/10.1021/jp1012973` (PMID: 20701382) doi: 10.1021/jp1012973

Zemla, J., & Austerweil, J. (n.d.). Analyzing knowledge retrieval impairments associated with alzheimer's disease using network analyses. *Complexity, vol. 2019*. Retrieved from `https://doi.org/10.1155/2019/4203158`