MIT 18.996: Topic in TCS: Internet Research Problems **Spring 2002**

Lecture 15 — May 15, 2002

Lecturer: G Plaxton *Scribes: R Fan, L Mccann, A Qureshi, Z Syed*

15.1 Network Visualization

The development of visualizations suited for communication networks is hardly a new phenomenon. The state in which data is presented to the human eye has a powerful effect on how the structure in those data is perceived, and it is for this very reason that visualization of large communications systems is critical to developing an understanding of their functioning. In particular, it represents an important problem, which has been tackled exhaustively in the past for certain media such as the telephone (for example, as early on as 1945, Cable and Wireless had developed techniques for modelling their network of links extending across much of the globe as shown in Figure 15.1) and is currently a subject of great interest to people interested in analysing the performance of the Internet.

This section of the lecture examines some of the contemporary methods used to visualize computer networks. More information can be found at *www.cybergeography.org* which contains links to numerous internet-related visualization projects.

15.1.1 Geographic Distance

Whenever nodes of a graph must be displayed they need to be arranged in space in some way. When we consider network hosts as the nodes, there are two important ways we can derive the spatial arrangement of those nodes. The first possibility is to arrange the nodes according to their geographic positions in the real world. Visualizations based on this approach generally display the hosts located on a map. Such visualizations tend to have a somewhat intuitive appeal to humans since these arrangements relate closely to the perceptions of the world held by most people.

The approach adopted by the Abilene Networks Operations Center at Indiana University to monitor communication between different routers, backbone links and customers at locations spread across the continental United State follows the discussion just presented. More specifically, as can be seen in Figure 15.2 a 2D map of the country is provided with nodes placed at points corresponding to their physical locations. These are then connected by edges that may provide information related to how much data is passing through the nodes either by using some colour coding scheme (to represent different levels of throughput; the choice of colours shall be discussed later) or by varying the thickness of the inter-connecting lines. Some graphical user interfaces (for example Figure 15.3) also incorporate the additional functionality of being able to click on links to view more details such as the traffic in either direction. Moreover, in some cases such as the one in Figure 15.4, it is also possible to zoom in on certain nodes using a non-linear fish-eye lens. This magnification allows users

Figure 15.1. Cable and Wireless (1945)

to focus on certain smaller portions of the visualization that are of more direct interest to them.

It is possible to refine visualizations based on geography even further. As Figure 15.5 shows, network activity can be presented in three dimensions rather than two, and a direct effect of this approach is that the traditional usage of edges can be augmented by the presence of sky-scraper like structures associated with each node, the height of which is related to the aggregate traffic processed at each point in the network.

It is important to note however, that although a cursory glance may fool observers into disparaging the problem of developing a means of visualizing what goes on at the network level as being relatively trivial, a deeper analysis readily dispels this myth. For example, consider the diagram showing UUNet's North America Internet Network provided in Figure 15.6. Although the map is still able to provide some meaningful information despite the jumbling of lines of various sizes (which represent the bandwidth of links between nodes) it is impossible to imagine that this will remain the case once we overlay all the other networks existing on the Internet atop this one. This will add tremendous confusion to the visualization (since the number of displayed nodes and edges will increase by at least an order of magnitude), making it difficult to glean any meaningful information from it at all.

There are various methods for reducing visual complexity when displaying the network. The sections that follow discuss some of them.

Figure 15.3. Multi Router Traffic Grapher (Oetiker)

Figure 15.4. Example of nonlinear magnification (Keahey)

Figure 15.5. Cox, Eick, and He (Bell Labs, 1993)

Figure 15.6. UUNET 2000

Figure 15.7. Global Internet map (TeleGeography, 2001)

Flow Aggregation Some visualization techniques such as those shown in Figure 15.7 aggregate flows between different points. In this case all the network links from one country to another are represented by one edge, the thickness of which is proportional to the quantity of data transferred. This approach reduces visual complexity by reducing the amount of information presented to the user: now only inter-country relations can be seen.

Node Clustering A further extension of the principal of aggregation is that of clustering nodes by region. In other words, rather than reducing complexity by attempting to decrease the number of edges present on a map, the same effect can be achieved by combining one or more nodes together through various clustering algorithms. This is particularly useful when the information users are interested in is the communication taking place between groups of nodes, rather than each individual point on a network. Figure 15.8 provides an example of this technique being adopted by Akamai. The second part of this lecture looks at some clustering algorithms.

Hierarchial Displays Another interesting way of presenting the massive amount of information related to network activity is by aggregating it in hierarchical layers. More specifically, the complexity of the visualization is drastically reduced by pursuing a strategy whereby we display backbones on one layer, children links with smaller bandwidth on the

Figure 15.8. GeoGraph (Akamai)

next and so on. As shown in Figure 15.9, this process drastically reduces the concentration of edges, thereby providing a more organized and hence clearer presentation of information. Unlike node clustering or flow aggregation, this approach does not reduce the information presented to the user; it simply organizes the information so that users are not overwhelmed by the complexity of the network topology.

15.1.2 Network Distance

The methods for monitoring network topology and activity mentioned uptil now suffer increasingly from the problem that they are rooted firmly in geography and do not pay heed to the fact that network distances are often radically different from their geographic counterparts. The role of physical distance is changing as network performance is complicated by variable speed network links, dramatically distorting the effect of physical distance. Two computers located nearby in the real-world may be 'far' apart from the perspective of a data packet that must move between them.

Automatic Graph Arrangement Network distances, usually measured by the hop-count or the average round-trip delay, therefore provide another method of arranging nodes for display. Automatic graph arrangement algorithms, such as spring-embedder variants, can be employed to derive an arrangement of nodes to maximize some given metric, where edge lengths represent network distances. The visualization produced by Burt and Cheswick in (Figure 15.10) follows this approach. Burch and Cheswick used `traceroute` from Internet hosts to collect data about the network's topology and then applied automatic graph arrangement algorithm's to this data.

Figure 15.9. Cox and Patterson (NCSA, 1994)

This form of network visualization is quite common and since there are many metrics that graph arrangement algorithms can depend on (number of edge crossings, average proximity to graph center, etc), there is a wide variety of visualizations which can be produced. Many projects at the Cooperative Association for Internet Data Analysis (www.caida.org) have adopted this approach.

Hyperbolic Trees Hyperbolic trees are a dynamic representation of hierarchical structure of the network and are an effective way to display complex trees clearly. One of the concepts behind this kind of representation is the varying level of detail, depending on the user interest: the objects the user is interested in, are displayed with full detail and centered on screen, while the others are put to the side, with reduced detail. Figure 15.11 shows an example of a hyperbolic tree.

The concept of hyperbolic trees can be extended to three-dimensional images, allowing users more control over the level of detail visible to them (rotation is also possible). Figure 15.12 shows a 3D hyperbolic tree view of a computer network.

SmartMoney Figure 15.13 shows another way of factoring information about 'throughput' (in this case, the volume of market shares) into the display. Each of several hundred companies in the market is represented by a rectangle with an area proportional to the compnay's market capitalization. Companies are grouped into categories (Technology, Financial, etc). The display is generated by using a tree to store company information sorted in such a

Figure 15.10. Burch and Cheswick (Lucent, 2000)

Figure 15.12. Hyun's Walrus network visualization tool (CAIDA, 2000)

Figure 15.13. Map of the Market (SmartMoney.com, 1998)

way that the given display can be generated by recursively processing the tree. For example, for Figure 15.13, the root of this tree could have as its left sub-tree the tree containing the categories in the left part of the figure (Technology, Financials, etc), and have as its right subtree the tree containing the categories in the right part of the figure (Energy, Capital Goods, etc). At the top level, the algorithm would determine the market share for each of these subtrees and draw a line dividing the display into two regions so that each subtree has the correct proportion of the display. It would then repeat the process on each of the two subtrees, dividing the display further at each stage.

15.1.3 Arranging Nodes: Hybrid Approaches

Both the geographic and the network distance approaches have their strengths; they both provide important information related to the network. The geographic view allows human observers to relate the network's topology to their world-view, while the network distance view, in some sense, presents an image of the network which highlights certain non-geographic aspects of its topology. A hybrid approach promises to combine the intuitive appeal of the former approach with the informative value of the latter. Interoute (Figure 15.14) applies a subway analogy to the internet; distorting the geography to accentuate network topology information. Skitter (Figure 15.1.3) also tries to derive an image based both on geographic as well as network connectivity factors. The sections that follow discuss various approaches to combining network information with geographic information.

Figure 15.14. Interoute I-21 network (Interoute)

Figure 15.15. The AS Internet graph: A Macroscopic Visualisation of the Internet During October, 2000.

Skitter Skitter presents an interesting approach to network visualization by combining geographic information with network topology features to produce the image shown in Figure 15.15. Skitter visualizes the Internet as a network of Autonomous Systems (ASes) with peering sessions being displayed as edges between the ASes.

Skitter maps each IP address to the AS responsible for routing it, i.e., the origin (end-of-path) AS for the best match IP prefix of this address in Border Gateway Protocol (BGP). The visualization displayed in Figure 15.15 uses routing tables collected by the University of Oregon's RouteViews project (`http://www.antc.uoregon.edu/route-views/`). The graph consists of 7,563 Autonomous Systems (81% of all ASes present in the Oregon BGP table on Oct. 15, 2000) and 25,005 peering sessions.

The position of each AS node is plotted in polar coordinates, position (radius, angle) or pos (r, θ) calculated using the equations in Figure 15.16. The outdegree of an AS node is the number of 'next-hop' ASes that were observed accepting traffic from this AS. The link color reflects outdegree, from lowest (blue) to highest (yellow),

Graphing dimensions of peering richness and geographic information reveals the highly

Figure 15.17. Presidential election results (NBC, 2000)

Figure 15.19. SOM of welfare of countries (Kaski 1997)

Kaski [1] gives a methodology of using SOMs for exploratory data analysis. Using principal component analysis and the idea of SOMs, informative network visualizations can be derived from the multidimensional analysis of network data.

Principal Component Analysis (PCA) can be used to display the data as a linear projection on a subspace of the original data space that best preserves the variance in the data. PCA is a standard method in data analysis; it is well understood, and effective algorithms exist for computing the projection.

Figure 15.19 shows a dataset projected linearly onto the two-dimensional subspace obtained with PCA. Each 39-dimensional data item describes different aspects of the welfare and poverty of one country. The data set consisting of 77 countries was picked up from the World Development Report published by the World Bank (1992). Missing data values were neglected when computing the principal components, and zeroed when forming the projections.

15.1.5 Choosing Colours

Most network visualizations use colours to represent additional information that is not apparent in the topology of the network or the arrangement or size of the nodes. For example, different colours are commonly used to represents different levels of data throughput. When colouring network visualizations most programmers choose the so called rainbow colour-map since there exists a simple mapping from numerical values to colours for this approach. Unfortunately, this mapping may produce misleading images.

Bergman et al. [2] at IBM have presented alternative approaches to colour-maps. The discussion below draws largely from their work.

In order to accurately represent continuous data, the visual dimension chosen must appear continuous to the user. An inappropriate colour-map, however, could create a visual representation which does not look monotonically increasing, the rainbow colormap is an example. How humans perceive colour and what ranges of the data need to be highlighted should, therefore, be the important factors determining the colours used.

Ensuring that continuous variables are mapped onto perceptually continuous dimensions, however, will only give a faithful representation of the structure of the data if the spatial characteristics of the representation are taken into account. Human ability to resolve spatial variations differs for the hue and luminance mechanism in human vision. The luminance mechanism is tuned to higher spatial frequencies (that is, high resolution, finely detailed, or small-grained features). Colormaps which include a luminance component, therefore, can adequately represent high-spatial-frequency information. The hue mechanism is tuned to lower spatial frequencies. Thus, saturation-based colormaps, which display variations in the magnitude of a hue, would be inadequate for conveying high spatial frequency information, but well-suited for representing larger-scale spatial variations.

The authors present a program, PRAVDAColor, which allows users to generate colour maps using the considerations given above. PRAVDAColor allows the user to identify ranges of data to highlight perceptually. Figure 15.20 and Figure 15.21 show how the colour maps can influence views of the same data. The image details are much more apparent to the human eye in Figure 15.21 because of the colour map.

15.2 An Akamai Network Visualization

15.2.1 Description

A visualization for representing the distribution of Akamai servers and customers was presented in the slide portion of the lecture. The user gets to specify the number of points that Akamai is to be collapsed to. The problem can be abstracted to the best location to place n points to represent all of the Akamai servers. There is a metric space with distance have the properties that it is symmetric, non-negative, and obeys the triangle inequality.

Figure 15.21. Data Explorer Visual Program Incorporating PRAVDAColor.

15.2.2 K-center and K-median Problems

There are two metrics that are commonly used to solve this problem are k-medians and k-centers. The k-median problem is solved by identifying k end points to minimize the sum of distances (service costs) to selected point. The k-center problem is solved by minimizing the maximum travel distance for any one point. This makes k-center sensitive to outliers.

The problem is that both problems are NP-hard. The k-center solution can trivially be approximated with a factor of two via a simple greedy algorithm. Such an algorithm is the Farthest Neighbor algorithm, known in 1986 due to Hochbaum and Shmoys. The algorithm is to pick arbitrary point as first point. Then choose the point farthest away from what was first chosen. Then choose the next farthest away point, etc.

The first constant factor approximation algorithm for k-median was in STOC (1999) by Charikar, Guna, Shmoys, and Tardos. The best result for the k-median is a factor of $(3+\epsilon)$ - Arya et al. STOC (2001). There is a lower bound on the best approximation for the k-median. The solution cannot be approximated to better than $(1 + \frac{2}{\epsilon})$ factor unless NP is a subset of DTIME $(n^{O}(lglgn))$.

Another problem of interest is to solve the k-center problem wile maintaining a hierarchical clustering. A factor of 8 approximation algorithm for this was proposed by Sanjay Dasgupta of AT & T.

15.3 Clustering

15.3.1 Hierarchical Clustering

Hierarchical clustering clusters nodes by successively breaking clusters into smaller clusters. For example, given a set of items $\{A, B, C, D, E\}$, one way to hierarchically cluster the items is as follows:

In this notation, the first cluster is $\{A, B, C, D, E\}$. This is broken into two clusters $\{A, B, C\}$ and $\{D, E\}$. Then $\{A, B, C\}$ is broken into $\{A, B\}$ and $\{C\}$, and so on. We would like a data structure for representing the clusters, and the order in which they were broken up. One such data structure is a *dendrogram*. A dendrogram resembles a tree. The root of the dendrogram is a cluster containing all the original items. Each time a cluster is broken up, a branch is formed which is lower than all previous branches. An example helps to illustrate this. Figure 15.23 shows the dendrogram for the clustering given in Figure 15.22. Notice that the highest branch (horizontal line) breaks the set into two clusters, $\{A, B, C\}$ and $\{D, E\}$. The next highest branch breaks cluster $\{A, B, C\}$ into clusters $\{A, B\}$ and $\{C\}$, and so on. Note also that we can find the stage in the hierarchy when there are exactly k clusters by finding a height on the dendrogram when there are exactly k vertical lines. For example, to find the stage when there are 2 clusters, we can look at the height indicated by the blue dashed line. This line crosses 2 vertical lines of the dendrogram, and gives us the clustering $\{A, B, C\}$ and $\{D, E\}$. Similarly, we can find the stage with 3 clusters by looking at the red dashed line. We have seen how to compactly represent a hierarchical clustering. But how do we obtain such a clustering in the first place? Since hierarchical clustering is a restricted class of clustering, it is not clear that there always

exists a good hierarchical clustering for problems like k-center or k-median. We first discuss some heuristic hierarchical clustering algorithms which are often used in practice. They all come under the category of *agglomerative clustering*, in which we start by treating the points as singleton clusters, and successively merge the "closest" clusters. The induced clustering is the merging process in reverse. The most popular agglomerative clustering algorithms are *single-linkage*, *complete linkage*, and *average-linkage*. They differ in their definition of "closest". Single-linkage clustering defines the distance between two clusters as the minimum distance between points in the clusters. Complete-linkage defines the distance as the maximum distance between the points. Average-linkage usually defines the distance as the average distance between points. These algorithms are popular because they have the fast running time of $O(n^2)$, where n is the number of points. However, Dasgupta showed in [3] that the linkage algorithms do not give constant factor approximations to the k-center problem in the worst case. Also in [3] however, a hierarchical clustering algorithm is introduced which gives a factor 8 approximation for k-center. This algorithm works by combining the *farthest-first traversal* algorithm of Hochbaum and Shmoys [4], which gives a non-hierarchical 2-approximate clustering, with a method for assigning points to centers which increases the cost of a clustering by a factor of at most 4.

15.3.2 Incremental Clustering

A relatively new type of clustering problem is *incremental clustering*. Incremental clustering is the online version of the traditional clustering problem, in which we imagine that we do not know all the points in advance, but must cluster them as they arrive. We restrict the amount of work we are allowed to do for each new point. In one version of the problem, we preset a threshold k. For each new point that arrives, we can either put the point into an existing cluster, create a cluster containing only the new point, or merge two existing clusters and then do one of the two previous operations. However, we require that all times, there are at most k clusters. Charikar and Panigrahy [5] showed that there exists no incremental clustering which can approximate the k-median problem to a constant factor. However, they give an constant factor approximation algorithm to k-median when we are allowed to keep up to $O(k)$ clusters at all times.

$$
\begin{array}{l}
ABCDE \\
ABC|DE \\
AB|C|DE \\
A|B|C|DE \\
A|B|C|D|E
\end{array}
$$

Figure 15.22. A hierarchical clustering of $\{A, B, C, D, E\}$

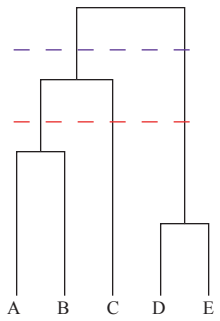

Figure 15.23. Dendrogram of a hierarchical clustering.

Bibliography

[1] Kaski, S. Data exploration using self-organizing maps. Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No. 82. DTech Thesis, Helsinki University of Technology, Finland. URL http://www.cis.hut.fi/ sami/thesis/.

[2] A Rule-based Tool for Assisting Colormap Selection, L. Bergman, B. Rogowitz and L. Treinish. Proceedings of the IEEE Computer Society Visualization '95 pp. 118-125, October 1995.

[3] S. Dasgupta. Performance guarantees for hierarchical clustering. In *Computational Learning Theory*, 2002.

[4] D. Hochbaum & D. Shmoys. A best possible heuristic for the k-center problem. In *Mathematics of Operations Research*, 10(2):180-184.

[5] M. Charikar & R. Panigrahy. Clustering to minimze the sum of cluster diameters. In *Proceedings of the 33rd Annual ACM Symposibum on Theory of Computing*, 2001.