ABSTRACT
Building and maintaining a reliable, high performance network infrastructure requires the ability of accurately visualizing, rapidly navigating and effectively resolving performance impacting issues. With the growing number of network entities and services, exploratory monitoring of a large-scale telecommunication network is becoming increasingly difficult. This paper presents a density hierarchy clustering algorithm, designed for real-time visualization of large telecommunications networks. The density histogram is calculated, which replaces the original dataset in further processing. The elements (cells) of the density histogram are compared to their neighbors in order to assign them to density hierarchies, which in turn identify the clusters. The experimental results have shown that the proposed algorithm provides high accuracy in visualizing node clusters, while significantly outperforming k-means in terms of clustering speed. This makes the algorithm a practical exploratory monitoring solution.

1. INTRODUCTION
One of the primary responsibilities of network operators and service providers is to ensure that their services provide a level of performance and robustness that satisfies the commitments specified in their service level agreements (SLAs) with customers. To achieve this, new network and service management systems are needed to continually monitor the quality and behavior of the services, identify unusual or anomalous activity that either directly indicates or indirectly implies that the service is no longer behaving satisfactorily, and analyze (root) causes of service performance degradations. Exploratory monitoring is one of these solutions.

Exploratory monitoring refers to those monitoring approaches that employ a variety of visualization techniques to maximize insight into a complex system, extract important variables, and detect outliers and anomalies. Exploratory monitoring is becoming increasingly important for understanding the structure of communication networks and services, as it is the most efficient method for a monitoring and management system to deliver to a human user the information pertaining to the structure and function of a telecommunications network.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

KDD4Service '11, San Diego, California, USA.
Copyright 2011 ACM 978-1-4503-0842-7 ...$10.00.

Nowadays, telecommunication networks have converged with traditional data communication networks like the Internet and sensor networks. The number of cells in an operational network has been increasing due to stringent service requirements. In parallel with the ever increasing network sizes has been a concomitant increase in collection of network data.

Under such scenarios, how to efficiently visualize the data associated with very large networks is becoming increasingly difficult. Exploratory monitoring solutions for large-scale networks may face the following challenges:

• Instant Interactivity. Visualizations must be interactive (e.g., visually interacting with the execution of a program), to efficiently convey the desired information to the user. It should be possible to quickly zoom-in and zoom-out, changing the view of the network from a network-wide view into a local network view. At different levels of zooming, the clustering of the data is different, in order to maintain a constant number of objects visualized on-screen. Thus, the zooming operations impose the requirement of a very efficient algorithm, capable of processing large datasets in real time.

• Accuracy. Each view shall reflect the characteristics of the network and the nodes (such as the distributions and densities of cells), to facilitate the correlations between network data and other information such as demographic data. Graphical displays, particular network maps, have long been recognized as a crucial tool for network analysis. Graphical displays, however, even when running on powerful workstation based systems, can become swamped and overly busy when faced with large networks and data volumes.

• Overhead. Monitoring of modern telecommunications networks produces vast amounts of data that need to be efficiently processed in order to extract useful information. The manipulation of the raw data, either by a human user, or an automated algorithm, can present severe difficulties, due to the sheer volume and diversity of the datasets. Processing of the raw data by an automated algorithm requires significant resources (time, computing power, memory), which may be scarce. Processing by a human user is also difficult, since displaying the data on-screen would overwhelm the user with the sheer amount of information.

The above are computational and usability issues respectively that are unfortunately synergistic, as one intensifies the other: the slow response of the computer would further hinder the (already difficult) conveying of information to the user.

This paper presents a density hierarchy clustering algorithm, designed for real-time visualization of large telecommunications networks. So far, the closest work we have found to this paper is
A 2-N across a 2-

density hierarchy clustering algorithm in detail. Section 3 presents a preliminary performance evaluation on the algorithm, which shows its advantages over one of the most popular clustering approach - k-means. Finally, conclusions are reached in Section 5.

2. BACKGROUND
Cluster Analysis is an open and very active research area with diverse applications like data mining, gene and protein analysis, image processing, among others. When clustering is used, the elements of the dataset are grouped together by similarity. These groups (clusters) of elements are processed, or visualized, instead of each element individually. This approach has the additional advantage of conserving the information of the dataset, while maintaining reachability with all the elements within it. Some of the existing algorithms are DBSCAN [3], k-means [4][5], OPTICS [6], BIRCH [7], DENCLUE [8], among many others. The frequency of those algorithms cited in the literature can be found in [9]. Two representative clustering algorithms are k-means and DBSCAN:

k-means is a simple and elegant algorithm, of both academic and practical interest. Performance is one of its main advantages [10], but its major drawback is that it requires a priori knowledge of the number and approximate location of the clusters. In addition, the production of clusters is sensitive to the initial centroids: poor initial choice may result in poor clustering results.

DBSCAN is a noise-resistant algorithm, capable of inferring the number of clusters, as well as detecting complex-shaped clusters, even concave-shaped. However, it may not respond well to data sets with varying density, an unavoidable feature in visualizing telecommunications networks. This observation is consistent with our own results, (Figure 1), where the production of clusters is often counter-intuitive. Furthermore, the sensitivity to the two parameters is demonstrated.

3. DENSITY HIERARCHY CLUSTERING
In this section, the proposed clustering algorithm is described in detail. It is shown how clusters are produced from a dataset comprised of N d-dimensional points p_n, n ∈ [1,N]. A 2-dimensional dataset is used as an example.

3.1 Density Histogram
Consider a distribution of N points p_n, n ∈ [1,N] across a 2-dimensional space, as shown in Figure 2a, with each p_n having coordinates x_n=x(p_n), y_n=y(p_n). It could be argued that the amount of entities hinder the conveying of information, which is further aggravated in larger and denser datasets.

The physical space where the p_n points lie is discretised by overlaying a rectangular grid of cells c_i, as depicted in Figure 2b. Each cell c_i is uniquely identified by a pair of coordinates (i,j) ∈ N^2, that specify its location within the grid, acting as coordinates in the respective axes (i for x, j for y).

Each point p_n in the original dataset corresponds to a single cell c_i in the grid, according to its coordinates x_n=x(p_n), y_n=y(p_n). The conversion from continuous (x,y) coordinates, to discrete (i,j) is described in (1),(2)^1.

\[
i = \left\lfloor \frac{(x_n - X_{\text{min}})X_{\text{res}}}{X_{\text{max}} - X_{\text{min}}} \right\rfloor (1) \quad j = \left\lfloor \frac{(y_n - Y_{\text{min}})Y_{\text{res}}}{Y_{\text{max}} - Y_{\text{min}}} \right\rfloor (2)
\]

Using this correspondence, each point p_n is assigned to its corresponding cell c_i.

There is a direct correspondence between the physical space and the grid of cells. The bounds of the physical space \{X_{\text{min}},X_{\text{max}},Y_{\text{min}},Y_{\text{max}}\} correspond to the bounds of the grid [1,

---

^1The floor function has been used for the conversion to an integer, but the ceiling function could be equally used, depending on the implementation; whether in the particular language arrays start from 0 or 1.
Problem Formulation

X_{res}, Y_{res} \text{, where } X_{res}, Y_{res} \text{ are the resolutions of the grid in the each axis, respectively. Thus } 1 \leq i \leq X_{res} \text{ and } 1 \leq j \leq Y_{res}.

The most important information that each cell c_{ij} holds is the amount of points (population) contained within it, denoted as d_{ij}. Since all cells are equal in the physical space, d_{ij} also carries the notion of density. Thus the density histogram is produced, which can also be interpreted as a discrete function of density. In this work, the terms density and population will be used interchangeably when referred to a cell. The time complexity of Algorithm 1 is O(N), as described in Algorithm 1.

Algorithm 1 Density histogram calculation

for all i,j : initialize d_{ij} to 0
for all elements p_n in the Dataset :
    calculate i from x_n=x(p_n), using (1)
    calculate j from y_n=y(p_n), using (2)
    increase d_{ij} by 1

The output of this algorithm is visualized in Figure 2c. The density histogram provides an intuitive overview when visualized, but its main merit is that it can be implemented as an array, therefore providing information on the density of an area in O(1) steps, as opposed to a kd-tree, B-tree, R-tree, or its variants, that require O(logN) steps. This is done by using the coordinates (i,j) of a cell to access it, or (i±1,j±1) to access one of its neighboring cells. This results in some loss of accuracy (resolution of the grid), but in applications like visualization (among others) this can be afforded for the benefit of performance.

The resolutions of the density histogram (X_{res}, Y_{res}), are decided based on N (the cardinality of the dataset), as well as the aspect ratio of the bounding box, which is approximately preserved for the density histogram. For visualization purposes, 100x100 has been found to be a sufficient upper limit for X_{res} and Y_{res}.

The other important merit of the density histogram, is that it may replace the original dataset in further calculations, which is the method followed in this work. Given its construction, it is bound to have significantly less elements to iterate on, than the original dataset, or 8Ch<<N. This, as can be seen in the experimental results, greatly boosts performance. The clustering is performed along the same guidelines, on cells instead of points.

3.2 Density Hierarchies

Instead of directly grouping together points p_n to form clusters, cells c_{ij} are grouped together, carrying along the points that each cell contains, and that have been assigned to the particular cell during the previous step.

Cells are connected in a hierarchical manner, with the most populated cells occupying the top level of the hierarchies, and the least populated, the bottom level (empty cells are ignored). Each cell c_{ij} can have many children-cells (all of which are less populated than itself), but only one parent-cell (which is more populated). A cell hierarchy from a real dataset is shown in Figure 3. The cell hierarchy can be interpreted as a discretised gradient ascent, since the parenthood connection (from child-cell to parent-cell) underlines the greatest increase in density, in a given radius around the child-cell, similarly to a density gradient.

In this paper, maxima refers to the cells for which no denser (more populated) cell is found in a given neighborhood, consistent with their role as local maxima in the discrete density function. Consequently, these cells don’t have a parent-cell, and therefore occupy the top of the cell hierarchies.

Figure 3. Density Hierarchies: The lighter cells represent denser areas, while the darker represent sparser or even empty areas. The arrows are directed from child-cell to parent-cell, denoting the gradual increments in density, and eventually reaching the densest areas, the maxima. The maxima have no parent-cell, since they are the most populous cells in their vicinity.

In this paper, a cluster refers to a collection of cells whose parenthood relations conclude towards the same maximum. Thus, nearby cells whose parenthood relations end up in a different maximum, belong to different clusters. Maxima and clusters have a one-to-one relationship and, from an implementation point of view, cells and points belonging to a single cluster can be accessed through their associated maximum.

Algorithm 2 Density Hierarchies

Neighbourhood := MinimumNeighbourhood //usually 3x3
reset OldNeighbourhood //to account for searching in a “shell”
repeat
    reset ListofMaxima
    for each non-empty Cell C_D of the Density Histogram
        for each non-empty Cell C_S in the
            (Neighbourhood-OldNeighbourhood) of C_D, that
            satisfies Population(C_S) > Population(C_D)
                if ((ParentCell(C_D)=Null) or 
                Population(ParentCell(C_D)) < Population (C_S) ) 
                ParentCell(C_D) := C_S
                if (ParentCell(C_D) = Null), add(C_D) to ListofMaxima
            OldNeighbourhood := Neighbourhood
            increase Neighbourhood
            until ( (Neighbourhood = NeighborhoodThreshold) or 
            (ListofMaxima.Count < MaxNumberOfClusters) )

In order to produce cell hierarchies, each cell seeks in an expanding neighborhood around it (Figure 4) for a parent-cell. A parent cell must be denser than the child-cell (central cell in figure 4). If more than one cell in a given neighborhood satisfy this criterion, the densest among them will become the parent-cell. This choice was followed in order to better emulate the continuous density gradient inside the discrete density histogram. Short-range parenthood relations are preferred, since they are less computationally expensive (cells further away are not examined), and they provide better spatial accuracy for the density gradient.
The search stops once a parent-cell is found, or a threshold to the size of the neighborhood is reached. In the latter case, the cell is named a maximum, and added to the list of maxima. The method is detailed in Algorithm 2, and the neighborhood threshold is an important parameter that is discussed next.

3.3 Neighborhood Threshold
The maximum range for parenthood relationships between cells is expressed through the neighborhood threshold. Modifying this parameter can affect the produced clusters, as can be seen in Figure 5a, b. A larger neighborhood threshold enables longer-range parenthood relationships. As a consequence, cells that were maxima for a smaller neighborhood can now become children of more densely populated cells, which are found within the range of the newly enlarged neighborhood. Since each maximum corresponds to a single cluster, the number of clusters decreases with a larger neighborhood threshold. Or equivalently, clusters discovered for a small neighborhood threshold, are progressively assimilated into clusters with denser cells, as the neighborhood threshold increases.

For visualization purposes, the maximum allowed distance can be used to approximate the number of objects to be rendered on-screen that is optimal for the user.

4. PERFORMANCE EVALUATION
The proposed algorithm was implemented and evaluated on a dataset representative of a real network, containing over 270,000 reports from network entities.

4.1 Qualitative Performance
Figure 6 demonstrates a portion of the dataset, along with two of the produced clusters. The proposed method can detect clusters of irregular shapes. Furthermore, the parenthood relations are very similar to density gradients, and that information can be useful for animated visualization of clusters: The cluster outline can initially only include a maximum, and successively expand along the directions of children-cells, eventually covering the whole cluster. Interpolation would be used to smooth the right angles of the cells outline.

This method is also resistant to sparse noise. The noise can be filtered by rejecting isolated (childless) maxima. These maxima are detected in relatively sparse areas, where no greater concentration of points exists, and therefore are likely to be noise. A further parameter is the population of a maximum (in relation to other maxima). In some special cases, maxima might be rejected if their population is below a certain threshold. This is a theoretical assumption, and no such need was indicated during the performance evaluation.

4.2 Quantitative Performance
Considering real-world IT environments for visualization applications, we run the experiments on low-cost commodity hardware: an Acer laptop with an Intel Core Duo CPU (1.83 GHz) and 4GB memory.

The performance of the proposed algorithm is compared against that of k-means. As mentioned earlier, the experiments were run on a dataset comprised of over 270,000 points. In order to assess the performance of the algorithms on smaller sub-datasets, random bounding boxes were drawn on the dataset, and clustering was performed on them.

One major disadvantage of k-means is the requirement on prior knowledge of the number and approximate location of the cluster.

The search stops once a parent-cell is found, or a threshold to the size of the neighborhood is reached. In the latter case, the cell is named a maximum, and added to the list of maxima. The method is detailed in Algorithm 2, and the neighborhood threshold is an important parameter that is discussed next.

3.3 Neighborhood Threshold
The maximum range for parenthood relationships between cells is expressed through the neighborhood threshold. Modifying this parameter can affect the produced clusters, as can be seen in Figure 5a, b. A larger neighborhood threshold enables longer-range parenthood relationships. As a consequence, cells that were maxima for a smaller neighborhood can now become children of more densely populated cells, which are found within the range of the newly enlarged neighborhood. Since each maximum corresponds to a single cluster, the number of clusters decreases with a larger neighborhood threshold. Or equivalently, clusters discovered for a small neighborhood threshold, are progressively assimilated into clusters with denser cells, as the neighborhood threshold increases.

For visualization purposes, the maximum allowed distance can be used to approximate the number of objects to be rendered on-screen that is optimal for the user.

4. PERFORMANCE EVALUATION
The proposed algorithm was implemented and evaluated on a dataset representative of a real network, containing over 270,000 reports from network entities.

4.1 Qualitative Performance
Figure 6 demonstrates a portion of the dataset, along with two of the produced clusters. The proposed method can detect clusters of irregular shapes. Furthermore, the parenthood relations are very similar to density gradients, and that information can be useful for animated visualization of clusters: The cluster outline can initially only include a maximum, and successively expand along the directions of children-cells, eventually covering the whole cluster. Interpolation would be used to smooth the right angles of the cells outline.

This method is also resistant to sparse noise. The noise can be filtered by rejecting isolated (childless) maxima. These maxima are detected in relatively sparse areas, where no greater concentration of points exists, and therefore are likely to be noise. A further parameter is the population of a maximum (in relation to other maxima). In some special cases, maxima might be rejected if their population is below a certain threshold. This is a theoretical assumption, and no such need was indicated during the performance evaluation.

4.2 Quantitative Performance
Considering real-world IT environments for visualization applications, we run the experiments on low-cost commodity hardware: an Acer laptop with an Intel Core Duo CPU (1.83 GHz) and 4GB memory.

The performance of the proposed algorithm is compared against that of k-means. As mentioned earlier, the experiments were run on a dataset comprised of over 270,000 points. In order to assess the performance of the algorithms on smaller sub-datasets, random bounding boxes were drawn on the dataset, and clustering was performed on them.

One major disadvantage of k-means is the requirement on prior knowledge of the number and approximate location of the cluster.
These operations can be as simple as rebooting, or more complex, malfunctioning elements, or to a single malfunctioning element. The network, and perform operations either on the bulk of the user could quickly isolate the failed elements from the rest of the network. In the previous example, with failed network elements, provides an intuitive interface for the user to interact with the network. Furthermore, the representation of clusters of network elements immediately after the failures have occurred. This manner, the density histogram does not have to be recalculated from the initial centroids. That makes k-means more comparable to the proposed method, since the same number of clusters is produced. Furthermore, given the varying density, with most point being very close to the maxima, k-means is expected to converge in very few iterations. Indeed, for over 500 different bounding boxes, the average number of iterations needed for k-means do converge is 8.72, with the standard deviation 5.93. The average number of clusters (also important for the performance of k-means) is 27.3 and the standard deviation 18.2. Figure 7 summarizes the results of the experiments.

The Neighborhood Threshold was set to 11x11 for all bounding boxes, and the resolutions of the density histogram (Xres, Yres) were calculated according to the number of points contained in each bounding box, bounded to 100x100.

As can be seen from Figure 7, the proposed algorithm significantly outperforms k-means, which makes it suitable for real-time data processing and visualizations in network monitoring systems. Such performance is critical given the dynamic nature of incoming information from the network. For example, in the case of some network elements failing, it is critical that this information is processed and presented to the user very quickly, so that appropriate actions can be taken.

4.3 Analysis

In this approach, dynamic data is handled very efficiently: the density histogram does not have to be re-calculated from the beginning, but only updated to account for the elements that have changed state (e.g. from functional to failure). In this manner, the cluster of the failed elements are presented to the user almost immediately after the failures have occur.

Furthermore, the representation of clusters of network elements provides an intuitive interface for the user to interact with the network. In the previous example, with failed network elements, the user could quickly isolate the failed elements from the rest of the network, and perform operations either on the bulk of the malfunctioning elements, or to a single malfunctioning element. These operations can be as simple as rebooting, or more complex like Root Cause Analysis (RCA), or even to dispatch a team for on-site repairs.

5. CONCLUSIONS

Network monitoring produces vast amount of data, neither feasible nor intuitive to visualize directly. If a network operator is presented with clusters of the network elements, instead of the elements themselves, he/she would have a much more intuitive picture of the network.

The proposed algorithm is designed for this purpose. The algorithm pinpoints the location of the cluster locations, allowing the subsequent algorithm to converge in very few steps, greatly boosting the performance; density histogram can provide with a very intuitive picture of the dataset, much more so than simply plotting the data according to their Longitude and Latitude.

Compared to k-means, this algorithm not only runs significantly faster, but eliminates the need of knowing a priori how many clusters should be formed.

6. ACKNOWLEDGMENTS

The authors would like to thank William Leahy, Piotr Stasiewski, and Jason Roche at Ericsson for their support with this work.

7. REFERENCES


