Deriving Cellular Network Structure From Inferred Handovers in a Cellular Association Trace

Brenton Walker
SICS Swedish ICT
Kista, Sweden
brenton.d.walker@gmail.com

Anders Lindgren
SICS Swedish ICT
Kista, Sweden
andersl@sics.se

ABSTRACT

A cellular association trace consists of timestamped events recording user activity in labeled cells in a cellular network. From such data one can infer that if a user appears in two different cells within a short span of time, that a handover took place, and that the coverage areas of the two cells overlap. That is, one can infer geographic information from handover behavior. One would like to expand this kind of inference to a larger scale, perhaps reconstructing a proximity graph of the cellular sites, or creating an approximate 2-dimensional embedding of the cells. We have analyzed a large-scale cellular association trace of several months of activity for several million users on a 3G network, and have found that handover behavior is actually incredibly diverse and complicated, making it very difficult to make any sort of global inferences, even in small sections of a network. In this paper we present some stable elements of handover behavior, and present several methods one can use to extract proximity information from such a trace.

Categories and Subject Descriptors

C.2.1 [Computer-communication networks]: Network Architecture and Design—*Wireless communication*

Keywords

cellular association trace; inferred handover

1. INTRODUCTION

There are several reasons one would want to study cellular association traces. In the field of mobile networking, researchers often want to build mobility models to drive simulations or validate models, for example [3, 8]. Doing this coherently can require understanding the arrangement and structure of the network, both for validation and semantic understanding. For example knowing that one group of cells covers a business district, and another group of cells is in a

small town would be useful for understanding and deriving meaning from an analysis.

One might also study such a trace for demographic or urban planning purposes. The associations of cell users can give us information about commuters, travel times, and cellular access patterns. Finally cellular carriers are interested in analyzing association traces to ensure that their network is functioning in an efficient and coherent way. Even with cell location information, studying the logical structure of handovers in the network can be valuable. Radio coverage is often modeled as circular disks, or a conic section. However the reality, involving interference, terrain, and buildings, is much more complicated, and very expensive and time-consuming to model accurately. When a handover takes place, the user can be taken as a *witness* to a specific point of overlap of the cells' coverage areas. Comparing predicted vs actual handover patterns could identify areas of poor coverage or excessive interference. Studying actual handover behavior is a step towards creating such models.

In this paper we document our work on the problem of constructing a cell proximity graph, and a corresponding 2D embedding, from inferred handover information. Our analysis is based on a cellular association trace for a 3G network, several months in length covering a full country with several million users appearing. We also have GPS coordinates for most of the cells, and use this to guide and evaluate our analysis. We present some statistics on the relationship between handover volume and actual cell distance, and present several possible criteria for adding edges to a cell proximity graph. We find the most important first step to this is grouping sets of co-located cells into combined *multi-cell* sites, and we present statistical properties of the handover data that show this is reasonable to do. Then we evaluate our ability to construct a handover-based proximity graph of the cell sites by comparing to the Delaunay triangulation of the sites, and show an example of using such a graph to compute a planar embedding of the cell sites.

Mobility/handover mechanisms and management are central to the functioning of cellular networks, and one can find a great deal of work proposing and analyzing handover models using different network and user mobility models [7, 2]. We have also seen work on performing node location estimation in a wireless network based on signal strengths or time-of-flight measurements [13, 9]. However we have not found any other work on using cellular handover information to infer geographic network structure.

Figure 1: The maximum and mean number of handovers between cells at different inter-cell distances.

2. ASSOCIATION TRACE DESCRIPTION

We have been given access to a large dataset of association traces from a large cellular operator. Our cellular association trace is a type of transaction data. Each data point consists of:

timestamp	userID	cellID	activity vector
-----------	--------	--------	-----------------

The **userID** is an anonymized hash. The **cellID** for 3G BTSes is a set of hierarchical identifiers. At the top level there is a Mobile Country Code (MCC) and Mobile Network Code (MNC). These are the same for the entire trace we use here. Within an MNC cells are grouped together by a Location Area Code (LAC) identifier. Within a Location Area each individual cell corresponds to a Service Area Code. So the full cell ID we refer to here is the Service Area Identifier (SAI):

$$\text{SAI} = \text{MCC} + \text{MNC} + \text{LAC} + \text{SAC}$$

Documentation [6, 1] indicates that in a UMTS network a single SAC may map to multiple BTSes, but that appears not to be the case in our trace. In fact there are usually many distinct SACs at every cell site.

The **activity vector** lists several attributes of the user's activity such as number of calls and SMS messages, and amount of data transferred up and down. The data points are aggregate, reported at 5-minute intervals. If a user had activity in more than one cell during a time interval, then she will have a separate entry in the trace for every cell she was active in during the 5-minute window.

We also have access to location information for the base stations for most of the 3G cells in the network. This meta-data consists of a list of cell IDs (MCC:MNC:LAC:SAC) and corresponding GPS coordinates. The GPS coordinates are given with varying levels of precision, and presumably some amount of error. Even if a pair of cells are not labeled with the exact same GPS coordinates, we consider a pair of cells to be co-located if their coordinates are within 25m of each other. This adjustment to the location data is justified by the following observations. If we look at the

Figure 2: Histogram of distance to 1st and 2nd nearest neighbors of each cell. The plot is truncated to 10km, though the support extends out to 107km.

inter-cell distance distribution we have 1502 pairs of cells that have non-identical GPS coordinates and a distance less than 100m, but only 302 pairs of cells with distance between 100-200m, then 632 pairs of cells with distances 200-300m, and so on. This suggests that cells are not generally placed closer than 100m to each other unless they are co-located, and that the large number of inter-cell distances less than 100m is due to different levels of precision or error in the GPS coordinates.

2.1 General Trace Statistics

The trace covers approximately six months and contains on the order of 10,000,000 transactions per hour, with much more activity during the day than at night. Approximately 10,000,000 distinct user IDs appear over the full duration of the trace, and about 130,000 distinct 3G cell IDs appear.

We naturally expect that distant cells will have no handovers, and that cells in close proximity will have many handovers. Figure 1 shows the one-month maximum and mean number of handovers between cells at different inter-cell distances. The y-axis is logarithmic. The expected inverse relationship between cell distance and handover volume is clear, but the inter-cell distances for which cells still see handovers is much larger than we expected. In fact, if we look for extrema, in one exceptional case we infer nearly 400 handovers per month between a pair of cells 90km apart. In other cases there are up to 13 handovers per month between cells over 500km apart. These appear to be due to a phenomenon where a phone has been switched off for some time, and when re-attaching to the network its initial activity is sometimes attributed to the cell where it was last seen. We filter out some of these anomalies by discarding any handovers that are close in time to an Attach Attempt. Unfortunately Attach Attempts can sometimes be recorded in the trace over an hour after the phone relocated. By setting a threshold on the number of handovers between a pair of cells, we can still filter out most of these anomalies.

We might expect cells in rural areas to cover much larger areas, and in these cases such long-distance handovers may be reasonable. A key statistic about the cell locations that can help us evaluate this is the distribution of each cell's nearest neighbors. Figure 2 shows the distribution of distances to the 1st and 2nd nearest neighbors to each cell. This plot is dominated by cells in urban areas where nearest neighbors tend to be between 200-2000m apart. However if we inspect the distribution of cells in sparser regions in the network we see that in some areas a cell's nearest neighbor may be over 66km away. While we do not know for sure how large these cells' coverage areas can be, this inter-cell distribution suggests that we can legitimately expect to infer handovers between cells over 10km apart.

3. TRACE PROCESSING

The main steps in our processing are to filter out bad events, identify inferred handovers, and finally build a proximity graph based on the inferred handovers. The final step can be done in a wide variety of different ways.

3.1 Initial Processing

The first step of our processing is to extract cellular handover information. We do not have direct information about handovers, so we infer it from the information we do have. If a user appears in two different cells within a small window of time, we reason that those two cells must have some overlap in their coverage areas. Unfortunately the time resolution of our trace is very coarse; the activity information is reported at five-minute intervals. A user in a car or train can cover a large distance in that time. Our hope is that given the large volume of trace data, geographically closer cells will still tend to have more handovers, even though fast-moving users may cause us to infer handovers between non-adjacent cells.

First we apply an activity filter to the trace to remove events where a user's connection to the BTS was too poor to successfully communicate. In order to consider an event valid we require that a user either successfully place or receive a call, successfully send an SMS message, successfully perform a location update, or transfer (up or down) at least 1024 bytes of data. This filters out 12.8% of events. Then for each five-minute reporting period we pull out all instances where a user reports successful activity in more than one cell, and record a handover between each pair of cells the user appears in. We exclude cases where the user also registers an Attach Attempt for the reason explained in section 2.1. On average in each reporting interval 14.5% of the active users appear in more than one cell.

Cell breathing is a phenomenon in CDMA networks where the effective range of the BTS shrinks as the cell becomes more loaded. However this does not affect the proximity of the cells. If anything, it will cause handovers of stationary and slow-moving users who are in the overlap of cells' coverage areas, which is a good thing for our purposes.

3.2 Proximity Graph Edge Criteria

If we add an edge to the inferred handover cell proximity graph for every single inferred handover, we do a terrible job of distinguishing between nearby and far away cells. We need to set a condition for when to add an edge to the proximity graph. The simplest method is to simply set a **count threshold**, T_c , for the number of handovers a pair

Figure 3: The PMF and CMF of number of co-located cells at a site.

of cells has to have before we add an edge. For a pair of cells, X, Y , let $H_c(X, Y)$ be the number of inferred handovers between those two cells. Then the simplest proximity criterion is to add an edge, XY to the proximity graph if $H_c(X, Y) \geq T_c$. The handover count can also be used to add distances/weights to the edges of the graph. Since we were interested in embedding the proximity graph in the plane, we treated the edge values as distances, and created a metric that gave node pairs with more handovers smaller distances. The metric we used for our experiments is of the form:

$$d_c(X, Y) = 1 + \frac{K}{H_c(X, Y)} \quad (1)$$

where K is a constant. In our experience, weighting the edges does not change the results much.

One problem with the count threshold is that it does not take into account the different handover volumes different cells have. One solution is to normalize H_c relative to total handover volumes of the pair of cells involved. If X and Y are two cells with total numbers of handovers C_X and C_Y respectively, then we can quasi-normalize the handover count by computing

$$\tilde{H}_c(X, Y) = \frac{H_c(X, Y)}{C_X + C_Y} \quad (2)$$

We say this is “quasi-normalized” because the counts are scaled based on the cells' activity, but the $\tilde{H}_c$ at any cell will not sum to 1.

Another approach we have found more effective is the **ranked handover fraction** criterion. For each cell, X , we make a list of all other cells $\{Y_i\}$ with which it has handovers, sorted by the number of handovers. We then add edges from X to the top-ranked Y_i up to the point where the connected cells account for a fixed fraction, $0 < \sigma \leq 1$, of X 's handovers. This approach requires no normalization, and tends to produce adjacency graphs that are closer to the Delaunay triangulation of the actual locations of the cell sites.

4. GROUPING CO-LOCATED CELLS

One thing that surprised us, is that there are typically many more BTSes than we expected co-located at each site. We had expected that there would be three cells on each tower, each with a different azimuth covering a 120° plane angle. In fact the number of cells at each site is typically a multiple of three, but is more commonly six, nine, or twelve,

with some instances of sites containing over 20 cells. Figure 3 shows the PMF and CMF of the number of cells at each site. We speculate that the larger numbers of co-located cells are to add capacity and operate on different frequencies, but we really do not know.

We have found that combining the sets of co-located cells into *multi-cells* drastically improves the quality and usability of network proximity graphs constructed based on inferred handovers. Part of this is due to the 10-fold reduction in the number of cells we have to deal with. Even more it is due to the clique-like interconnectivity of co-located cells in the handover graph and irregular connectivity to the cells at neighboring sites. Therefore detecting groups of co-located cells is a useful first step in analyzing cellular network structure based on inferred handovers.

We expect that if a cell is co-located with other cells, it will have a lot of handovers to its co-located brothers. This turns out to be true. In fact we have found the fraction of handovers between a cell and the others at the same site is surprisingly stable at about 45%, and does not depend on the number of cells at the site. The table below shows the mean handover fraction from each cell to others at its site for all cells in a few of the largest municipalities in our trace, and overall.

Municipality	mean handover fraction	st. dev.
1	0.42	0.04
2	0.45	0.04
3	0.43	0.04
4	0.45	0.04
overall	0.43	0.04

However for each cell in a co-located group, the other cells at the site are not necessarily the top-ranked handover neighbors. For a site with 12 or more co-located cells, for example, the co-located handovers simply have to be split more ways, and the co-located cells are ranked even lower in the listing of handover neighbors. This means that using the ranked handover fraction criterion to identify co-located sets of cells will not work well. On the other hand, if we use a handover count threshold to form a proximity graph, co-located sets of cells do tend to *almost* form cliques. Figure 4 shows the average graph density (the number of edges divided by the number of edges there would be in a complete graph) of co-located sets of cells for different T_c thresholds in the 13 largest municipalities in our trace.

This means that we should be able to approximately group the cells into co-located sets using graph partitioning algorithms. Graph partitioning is a computationally hard problem to solve exactly, but creating heuristic algorithms to find approximate solutions is an established subfield of graph theory [4]. We do not go into its details here.

5. PROXIMITY GRAPH OF SITES

We still need to evaluate our ability to construct a proximity graph from inferred handover information. In this section we deal exclusively with aggregated *multi-cells* consisting of sets of co-located cells, and refer to these sets of cells as “sites”.

We would like our proximity graph to be representative of the actual proximity of the sites, and be a good input for graph arrangement algorithms such as Yifan Hu [5] and low-dimensional embedding algorithms such as Isomap [10] or maximum variance unfolding [11].

Figure 4: Average subgraph density of sets of co-located cells in 13 largest municipal areas for several different T_c thresholds.

Probably the most elegant triangulation of planar points is the Delaunay triangulation. It has a number of desirable properties, in particular, being a planar triangulation (and therefore a planar graph) and the fact that the circum-circle of any Delaunay triangle does not contain any other vertex in the graph. The Delaunay triangulation has been proposed as the basis for algorithms for managing handovers and mobility in cellular networks [12], so we see it as a sort of ideal for the proximity graph we would like to derive from our inferred handover information.

The two main types of defects our proximity graph can have are missing connections and short-circuits. If we do not have enough data, or if we set the threshold for adding edges too high, the handover-based proximity graph will not have enough edges to represent the network. Lacking edges in comparison to the Delaunay triangulation is not always a defect, though. The Delaunay triangulation triangulates the convex hull of a set of points. If the actual network is not convex, or has un-covered areas, the Delaunay triangulation will still triangulate those areas with extremely long edges, and will therefore contain edges that we would not want in our proximity graph.

Short-circuits in the proximity graph are much more problematic. These are cases where we add an edge between two sites when there are one or more other sites in between. A graph with many short-circuits is difficult to unfold/embed in the plane because the short-circuit edges ruin the graph distance metric. These sorts of short-circuits will inevitably occur in our graphs, however. For example, along the path of a highway, a lot of users may show activity on several fairly distant cells with or without hitting ones that are in between. This is a bit of information about the mobility in the network that we would like to keep, but too many short-circuits all throughout the network will make it impossible to extract any information about the network or user mobility.

We compare the handover-based proximity graphs we compute from the trace to the Delaunay triangulations we compute from the actual cell locations by looking at the shortest paths through the graph for all pairs of sites. If the shortest path between a pair of sites is longer in the proximity graph than in the Delaunay triangulation, it indicates miss-

Figure 5: The relative length of shortest paths, both longer and shorter than the corresponding shortest path in the Delaunay triangulation. The shorter paths indicate site pairs whose shortest paths have “short-circuits”, edges that jump over what ought to be intermediate sites.

ing edges, or a lack of connectivity in the proximity graph (which may not be a problem). If the shortest path in the proximity graph is shorter than in the Delaunay triangulation it indicates the presence of short circuits (which may be inevitable, but is always a problem for planar embedding). We measure these two types of errors separately.

Let $\mathbb{X} = \{X_1, X_2, \dots, X_n\}$ be our set of sites, and $\mathcal{H}_\sigma(\mathbb{X})$ be the handover-based proximity graph computed from the trace using a ranked handover fraction of σ , and let $\mathcal{DG}(\mathbb{X})$ be the Delaunay triangulation of those points. For any graph, $\mathcal{G}$, let $\phi(\mathcal{G}, X, Y)$ be the length of the shortest path between vertices X and Y . The individual errors in path length are scaled relative to the length of the path in the Delaunay triangulation.

$$\Phi^+ = \sum_{X, Y \in \mathbb{X}} \frac{\max(0, \phi(\mathcal{DG}(\mathbb{X}), X, Y) - \phi(\mathcal{H}_\sigma(\mathbb{X}), X, Y))}{\phi(\mathcal{DG}(\mathbb{X}), X, Y)}$$

$$\Phi^- = \sum_{X, Y \in \mathbb{X}} \frac{\min(0, \phi(\mathcal{DG}(\mathbb{X}), X, Y) - \phi(\mathcal{H}_\sigma(\mathbb{X}), X, Y))}{\phi(\mathcal{DG}(\mathbb{X}), X, Y)}$$

For a quantitative evaluation, we constructed handover-based proximity graphs for the cell sites in four of the largest municipalities in our trace, and compared them to the corresponding Delaunay triangulation using the error functions above. We used the ranked handover fraction criterion for edge inclusion with fraction $\sigma = 0.5$. The graphs that we generated also have distances/weights attached to each edge based on the $d_c(\cdot)$ function in equation 1. This allows us to filter the edges included in the graphs which we evaluate. Let T_d be the threshold we use to filter the edges in the handover-based proximity graph. Figure 5 shows the error functions computed for a range of values of T_d for the four large municipalities. Figure 6 shows the distribution of the absolute errors.

Figure 6: The distribution of the absolute discrepancy in length of shortest paths in the inferred proximity graph compared to the Delaunay triangulation for $T_d = 5$

Both Φ^+ and Φ^- stabilize to very consistent values, and this plateau actually continues out to about $T_d = 500$. In this region the proximity graphs are very stable with few new edges being added for increased T_d . This is a good thing because a researcher working without the ground truth of cell locations has a huge and obvious “sweet spot” for choosing the T_d threshold.

From figure 5 we see that of the pairs of sites that have short-circuits, the average length of the shortest path through the proximity graph is about 23 – 29% shorter than that through the Delaunay graph. Of the site pairs that have shortest paths longer than that through the Delaunay graph, the paths on average are 40% longer than the ideal. We note again that paths longer than in the Delaunay triangulation are not necessarily a problem. Though it is not represented in the graph we find that about 25% of site pairs have short-circuit paths, and about 50% have longer paths. The remaining pairs have shortest paths that match the length of the shortest paths in the Delaunay triangulation.

6. PLANAR EMBEDDING OF CELL SITES

Though it is a more subjective result to evaluate, we have used Isomap to compute planar embeddings of our handover-based proximity graphs for several large municipalities. There are a variety of tools for dimensionality reduction, but finding a low-dimensional embedding of our cell proximity graph is inherently non-linear. The data points have a metric, but it is almost certainly non-euclidean. We chose to experiment with Isomap [10] because it is well-used and designed to find embeddings for arbitrary metric data.

Let D_G be a graph edge distance matrix. Isomap uses classical Multi-Dimensional Scaling (MDS) to generate a d -dimensional embedding in Euclidean space that minimizes the cost function:

$$E = \|D_G - D_\gamma\| \quad (3)$$

Here D_γ is the distance matrix of the resulting Euclidean embedding. In many applications involving high-dimensional data the true dimension of the data is unknown, so one computes embeddings for several different dimensions and then plots the resulting residual variance to estimate the correct

Figure 7: A planar embedding computed by Isomap from the handover-based proximity graph of the cell sites in one of the larger municipalities in the trace. Geographically close sites are marked with similar colors.

embedding dimension. In our case we *expect* the embedding to be two-dimensional, so we focus just on those results.

Figure 7 shows an example embedding computed for one large municipality. Sites that are geographically close have similar colors. We see that the proximity graph we construct does a fairly good job of preserving the actual geographic proximity, though the network is rotated and distorted from the true geographic arrangement. However there is enough correct information about the relative arrangement and proximity of sites that one could use this embedding to visualize and model user mobility between the sites.

7. CONCLUSION AND FUTURE WORK

We have discussed some of the challenges involved in the apparently simple problem of extracting cellular network structure from a large trace of coarse cell association data. Performing such analysis could be useful for validating anonymized trace data, analyzing network behavior, building mobility models, and building handover models to help detect faults and irregular network behavior. Inferred handover data contains some amount of geographic information, but tying these bits of information together into a more global picture is challenging. We presented several criteria for adding edges to a handover-based proximity graph, and discussed which methods work better for different purposes. In particular using a plain count threshold works better for detecting co-located cells, while using a ranked handover fraction filter works better for constructing a proximity graph on the co-located cell sites. We presented and used a technique to quantify the quality of the proximity graph we produce. Finally we summarized how we used the Isomap nonlinear embedding algorithm to construct some example embeddings of the cell sites of the largest municipalities in the trace.

8. ACKNOWLEDGMENTS

This work was carried out during the tenure of an ERCIM “Alain Bensoussan” Fellowship Programme.

This work was partially funded by the Future Networking Solutions action line of EIT Digital, by the FP7 Marie Curie IRSES project MobileCloud under grant agreement No. 612212, and by the KKS funded READY project.

9. REFERENCES

- [1] 3rd Generation Partnership Project. Technical specification group core network and terminals; numbering, addressing and identification (release 7).
- [2] W. Bao and B. Liang. Handoff rate analysis in heterogeneous cellular networks: A stochastic geometric approach. In *MSWiM '14*, pages 95–102, New York, NY, USA, 2014. ACM.
- [3] R. A. Becker, R. Caceres, K. Hanson, J. M. Loh, S. Urbanek, A. Varshavsky, and C. Volinsky. Route classification using cellular handoff patterns. In *Proceedings of the 13th International Conference on Ubiquitous Computing, UbiComp '11*, pages 123–132, New York, NY, USA, 2011. ACM.
- [4] C. Bichot and P. Siarry. *Graph Partitioning*. ISTE. Wiley, 2013.
- [5] Y. F. Hu. Efficient and high quality force-directed graph drawing. *The Mathematica Journal*, 10:37–71, 2005.
- [6] S. Kasera and N. Narang. *3G Mobile Networks*. McGraw-Hill Professional Engineering. McGraw-Hill, 2004.
- [7] R. Langar, N. Bouabdallah, and R. Boutaba. A comprehensive analysis of mobility management in mpls-based wireless access networks. *IEEE/ACM Trans. Netw.*, 16(4):918–931, Aug. 2008.
- [8] C. B. Merkebe Getachew Demissie, Gonalo Homem de Almeida Correia. Exploring cellular network handover information for urban mobility analysis. *Journal of Transport Geography*, 31(Complete):164–170, 2013.
- [9] M. Robinson and I. Psaromiligkos. Received signal strength based location estimation of a wireless lan client. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 4, pages 2350–2354 Vol. 4, March 2005.
- [10] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319, 2000.
- [11] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *Int. J. Comput. Vision*, 70(1):77–90, Oct. 2006.
- [12] Z. Yang, X. Liu, Z. Hu, and C. Yuan. Seamless service handoff based on delaunay triangulation for mobile cloud computing. *Wireless Personal Communications*, pages 1–15, 2014.
- [13] M. Youssef, A. Youssef, C. Rieger, U. Shankar, and A. Agrawala. Pinpoint: An asynchronous time-based location determination system. In *MobiSys '06*, pages 165–176, New York, NY, USA, 2006. ACM.