Towards Demographic Information Release in LBS K-Anonymization

by

Michael Andrew Sanchez

A Thesis Presented in Partial Fulfillment
of the Requirement for the Degree
Master of Science

Approved November 2014 by the
Graduate Supervisory Committee:

Gail-Joon Ahn, Chair
Adam Doupe
Partha Dasgupta
.

ARIZONA STATE UNIVERSITY

December 2014

ABSTRACT

The increasing number of continually connected mobile persons has created an environment conducive to real time user data gathering for many uses both public and private in nature. Publicly, one can envision no longer requiring a census to determine the demographic composition of the country and its sub regions. The information provided is vastly more up to date than that of a census and allows civil authorities to be more agile and preemptive with planning. Privately, advertisers take advantage of a persons stated opinions, demographics, and contextual (where and when) information in order to formulate and present pertinent offers.

Regardless of its use this information can be sensitive in nature and should therefore be under the control of the user. Currently, a user has little say in the manner that their information is processed once it has been released. An ad-hoc approach is currently in use, where the location based service providers each maintain their own policy over personal information usage.

In order to allow more user control over their personal information while still providing for targeted advertising, a systematic approach to the release of the information is needed. It is for that reason we propose a User-Centric Context Aware Spatiotemporal Anonymization framework. At its core the framework will unify the current spatiotemporal anonymization with that of traditional anonymization so that user specified anonymization requirement is met or exceeded while allowing for more demographic information to be released.

To all those that put up with me

TABLE OF CONTENTS

Page

LIST OF TABLES . . . v

LIST OF FIGURES . . . vi

CHAPTER

1 INTRODUCTION . . . 1

2 BACKGROUND . . . 5

2.1 NBLS k-Anonymization . . . 5

2.1.1 Definitions . . . 5

2.2 LBS k-Anonymization . . . 8

2.2.1 Definitions . . . 9

2.2.2 Related LBS K-anonymization . . . 11

2.3 Unaddressed in Current Techniques . . . 13

3 FRAMEWORK FOR K-ANONYMOUS M-COMMERCE . . . 18

3.1 Anonymization Service . . . 19

3.1.1 Possible NLBS/LBS k-Anonymization Combinations . . . 19

3.1.2 Algorithm and Data Structures . . . 22

3.2 End-user . . . 39

3.3 LBS Provider . . . 40

4 IMPLEMENTATION . . . 42

5 EVALUATION . . . 46

5.1 Affects of Policy on Time . . . 46

5.1.1 Affects of Policy on Processing Time . . . 48

5.1.2 Affects of Policy on Waiting Time . . . 49

5.2 Affects of Policy on Information Loss . . . 50

5.2.1 Effects of Policy on NLBS Information Loss . . . 52

5.2.2 Effects of Policy on LBS Information Loss 54

5.3 Discussion 54

6 CONCLUSION 57

6.1 Future work 58

REFERENCES 60

APPENDIX

A DATA COLLECTED FROM EXPERIMENTATION ON SIMULATED USER BASE 63

LIST OF TABLES

Table Page

2.1 NBLS Raw User Information ... 7
2.2 NLBS k-Anonymized k=2 ... 7
2.3 NLBS k-Anonymized k=4 ... 8
2.4 NBLS Raw User Location Information ... 14
2.5 Combined Raw User Location Information ... 16
2.6 LBS k-anonymous resultant, k=4 ... 16
2.7 NLBS k-anonymous resultant, k=4 ... 17
3.1 NLBS k-Anonymized k=2 ... 21
3.2 LBS k-Anonymized k=2 Unsuccessful ... 21
3.3 Multiracial Modified NBLS Raw User Information ... 26
3.4 Multiracial Modified NBLS k=2 ... 28
A.1 Average NLBS Generalization Percentage ... 64
A.2 Average STD of NLBS Generalization Percentage ... 66
A.3 Average LBS Generalization Percentage ... 68
A.4 Average STD of LBS Generalization Percentage ... 70
A.5 Average Generalization Time ... 72
A.6 Average STD of Generalization Time ... 74

LIST OF FIGURES

Figure Page

2.1 Gender DGH and VGH . . . 6

2.2 Race DGH and VGH . . . 6

2.3 Origin DGH and VGH . . . 6

2.4 General Framework . . . 9

2.5 Coordinate DGH and VGH . . . 14

2.6 User locations . . . 15

3.1 Combination Possibilities . . . 19

4.1 Package Diagram . . . 42

4.2 Incoming Queries . . . 44

4.3 Outgoing Queries: part 1 . . . 44

4.4 Outgoing Queries: part 2 . . . 44

4.5 Global Configuration . . . 45

5.1 Average Turnaround Time (sec) . . . 47

5.2 Log Average Turnaround Time (sec) . . . 47

5.3 Average Processing Time (ms) . . . 48

5.4 Average Wait Time (sec) . . . 49

5.5 Log Average Wait Time (sec) . . . 50

5.6 Average Overall Percent Generalized . . . 51

5.7 Average NLBS Percent Generalized . . . 51

5.8 Average Age Percent Generalized . . . 52

5.9 Average Gender Percent Generalized . . . 52

5.10 Average Origin Percent Generalized . . . 53

5.11 Average Race Percent Generalized . . . 53

5.12 Average LBS Percent Generalized . . . 54

Figure Page

5.13 Average Latitude Percent Generalized 55

5.14 Average Longitude Percent Generalized 55

Chapter 1

INTRODUCTION

As of 2013 the number of adults in the United States with mobile phones hit 91% [15]. The capabilities of many of these devices go beyond phone calls. They are in fact a launching point for many different services including navigation, shopping, and social media. Each of these services requires differing types and amount of information that is as accurate as possible in order to provide a useful result. Navigation for example would require Global Positioning System GPS coordinates. While a social media application requires information such as name, gender, and age. However as the mobile platform has developed, services that require both navigation-esque and social media-esque information have begun to emerge.

Regardless of the information required by the service, privacy remains a concern of the mobile user. In Location-based Services LBS such as navigation the privacy concern is regarding a user's physical location at a given time. Release of this information allows an attacker to determine movement patterns of an user. Possibly leading to a determination of said user's home address, place of employment, establishments frequented, and typical routes taken. In Social media-esque $NLBS$services the privacy concern is release of demographic information intrinsic to but not directly identifying an user such as age, gender, ethnicity, and country of origin. This information can be combined with other publicly available data sets in a linking attack eroding the user's privacy [8]. Mitigating the risk of releasing information related to but not directly identifying an user while providing accurate information for services has borne the research area of k-Anonymity.

Conceptually k-Anonymity is hiding a user in a crowd of k-1 other individuals that are indistinguishable from said user. It utilizes generalization and suppression of quasi-identifiers to accomplish this end. Generalization decreases the accuracy of quasi-identifiers while suppression removes outliers from a the crowd. Quasi-identifiers are pieces of information which by themselves reveal little to nothing about a user but when coupled with other sources of information may reveal unexpected and/or private data of a user. Due to the nature of their required user information LBS and NLBS differ in their quasi-identifiers. LBS k-Anonymity focuses on spatiotemporal quasi-identifiers. While NLBS k-Anonymity handles demographic quasi-identifiers.

Originally NLBS k-Anonymization was designed for the release of user data from large databases. Later k-Anonymization was adapted for use in LBS by viewing the spatiotemporal point of a user a quasi-identifier. NLBS k-Anonymization is designed to handle relatively static data and produce an absolute minimal generalized data set for the entire initial set. In this paradigm an the initial user data set is considered complete and a minimal generalization is determined using the full information available. On the other hand LBS k-Anonymization is designed to process dynamic data and produce an absolute minimal generalized data set at a given instance. The initial user data set is incomplete and a minimal generalized data set is sought for every new piece of user data. In addition to these differences NLBS and LBS operate on different concepts of generalization; NLBS uses a fixed tree structure for generalizing quasi-identifiers while LBS utilizes a fixed function for generalizing quasi-identifiers. Given these differences between LBS and NLBS k-Anonymization providing user privacy for services which require both spatiotemporal and demographic information is a difficult task.

NLBS k-Anonymization uses a tree structure for anonymization where the leafs represent no generalization and the root is full generalization. With this structure there is only a single path of generalization for each leaf. While LBS uses a fixed search function where a single spatiotemporal point represents no generalization. The full generalization is represented by a maximum sized spatiotemporal region. The search function will yield different results for different initial user data sets. Therefore there exists many possible generalization paths for any given spatiotemporal point. In addition to these different generalization mechanisms NLBS and LBS k-Anonymization also differ in speed goals due to the nature of their use. Many LBS have a tight time limit to produce a result. For example in navigation a user reasonably expects their directions to match their current location which would not be possible if the LBS took longer than a few seconds to operate. NLBS is not expected to operate in a manner of seconds as the datasets they handle are large and timeliness of their resultant does not affect the use of the service. As NLBS was initially formulated to release data to researchers a day or longer of operating time is not unreasonable.

Achieving k-Anonymization for services with both LBS and NBLS user data requires features of both their respective k-Anonymization techniques. Neither LBS nor NBLS k-Anonymization is suitable for handling types of user information that the other handles. NBLS k-Anonymization will generally yield an over generalized spatiotemporal region. While LBS k-Anonymization requires a well-ordered set in order to generalize and user demographic generalization is a tree i.e. partial ordering.

In this thesis we present a k-anonymization approach that is independent from the underlying generalization structure. The approach uses a concept of *similarity* and *difference* in order to provide a minimal k-Anonymization for both NLBS and LBS types of quasi-identifiers. It operates in an LBS environment with time constraints and incomplete a priori knowledge of the user set. The resultant anonymized data set can be used by both advertising services and governmental agencies to best react to the composition of people in a given spatiotemporal area. We have implemented a prototype anonymization service as part of this research as well as simulated various user population compositions, sizes, and rates of movement.

The remainder of the thesis is structured as follows. Chapter 2 covers NLBS and LBS k-Anonymization background information and illustrates the need of a different approach with an example. Chapter 3 overviews the general framework of our approach and describes the anonymization service. Chapter 4 discusses the implementation details and generation of simulated user traces. Chapter 5 covers the evaluation of our approaches performance and discussion of limitations. Chapter 6 concludes the thesis and presents possible future directions of this work.

Chapter 2

BACKGROUND

The related works in spatiotemporal k-anonymity have focused on expanding, read generalizing, initial coordinates into a larger and larger spatiotemporal regions until some criteria are met while not degrading the QoS. Please note that the expansion pattern is not fixed in most cases, an initial spatiotemporal point may expand into any number of generalized results. Conversely k anonymity as presented in [8; 5; 22; 25**?** ; 3; 24; 2] uses a fixed anonymization structure.

2.1 NBLS k-Anonymization

Releasing *truthful* information for "circulation or research" [8] is the primary focus of NLBS k-Anonymization techniques. These techniques take in complete user data sets then transforms them into k-anonymous data sets via *generalization* and/or *suppression*.

2.1.1 Definitions

Definition 1 (k-anonymity)**.** Let $T(A_1, \ldots, A_m)$ be a table, and QI be a quasi-identifier associated with it. T is said to satisfy k-anonymity with respect to QI iff each sequence of values in T[QI] appears at least with k occurrences in T[QI].

As discussed previously *generalization* utilizes a tree structure to replace exact values of an quasi-identifier with more general versions. Suppression on the other hand will remove outliers from the original data set. Generalization may or may not be combined with suppression while suppression is normally paired with generalization

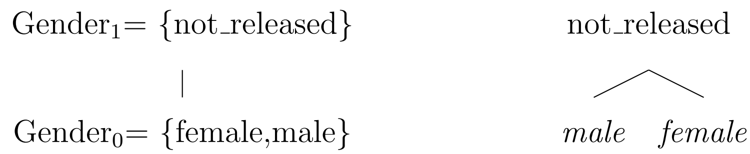

Figure 2.1: Gender DGH and VGH

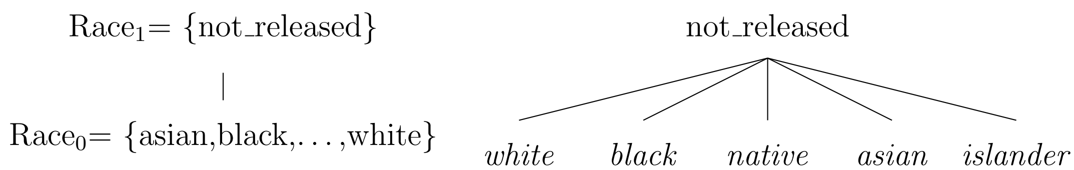

Figure 2.2: Race DGH and VGH

as pure suppressive approaches have "limited applicability" [8]. Figures 2.1.1 through 2.1.1 show example generalization hierarchies.

The Domain Generalization Hierarchy (DGH) is well-ordered set representing the levels of generalization possible for a given domain. While the Value Generalization Hierarchy (VGH) is a partially ordered set where all paths from root to a leaf have the same number of intermediate nodes. The VGH contains the actual values a quasi-identifier may assume at any given level. This representation works well for quasi-identifiers with no order amongst ungeneralized values. Take gender for example, though each sex may have their own views of superiority, there is no way to say male comes before female and vice versa. A lattice is used to represent more than one

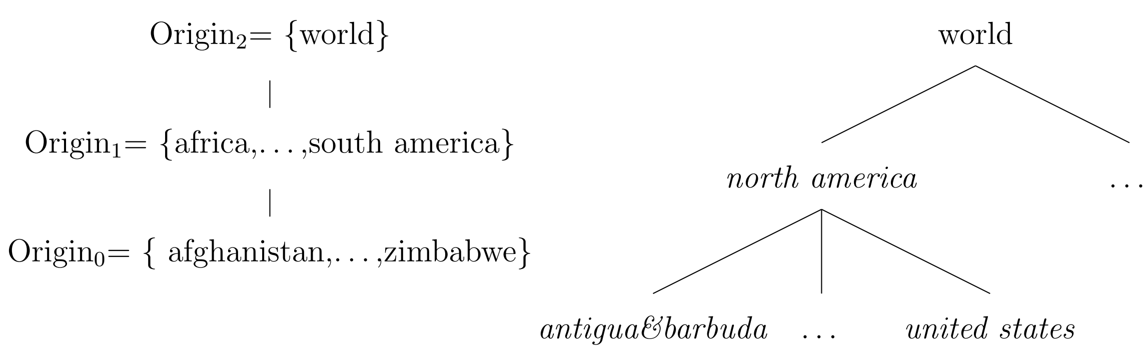

Figure 2.3: Origin DGH and VGH

quasi-identifier where every point is a composed of all quasi-identifiers and between adjacent nodes only a single component is changed. I.E. when a node is a single hop away all of its components will match the source node with the exception of a single component and the difference between the source node and destination node component is a single level of said components own DGH.

Name	Gender	Race	Origin	Content
Mike	male	white	United States	C_m
France	female	black	Haiti	C_f
Eusebio	male	white	Mexico	C_e
Tosh	female	native	United States	C_t
Nesto	female	asian	Mexico	C_n

Table 2.1: NBLS Raw User Information

Name	Gender	Race	Origin	Content
Mike	male	not_released	north america	C_m
France	female	not_released	north america	C_f
Eusebio	male	not_released	north america	C_e
Tosh	female	not_released	north america	C_t
Nesto	female	not_released	north america	C_n

Table 2.2: NLBS k-Anonymized k=2

An example of an NLBS k-Anonymized user data can be seen in tables 2.2 and 2.3 for k=2 and k=4 respectively. Given the user information in table 2.1 an anonymization service begins at the leafs of the VGH for each of quasi-identifiers in table 2.1 and traverses towards the root node until a generalization near the original leafs is located. As these algorithms function on entire data sets the generalization is applied

Name	Gender	Race	Origin	Content
Mike	not_released	not_released	north america	C_m
France	not_released	not_released	north america	C_f
Eusebio	not_released	not_released	north america	C_e
Tosh	not_released	not_released	north america	C_t
Nesto	not_released	not_released	north america	C_n

Table 2.3: NLBS k-Anonymized k=4

to all user's regardless of necessity. For $k = 2$ a generalized resultant satisfying the k requirement is at $Origin_1, Race_1, Gender_0$. Increasing the k requirement tends to reduce the amount of useful information that can be released as is shown in the table 2.3 which has a generalization located at $Origin_1, Race_1, Gender_1$. Table 2.2 releases gender and some origin information while Table 2.3 shows only origin information. Please note that items such as Name are considered direct identifiers, reveal the user directly, and would either be removed or replaced before releasing the k-anonymous table. As discussed above the generalization is applied to the entire data set this results in Mike and Eusebio race being *not_released* for $k = 2$ even though no generalization was required as they shared the same race. The race was withheld due to France, Tosh, and Nesto requiring a generalization of race to meet the k requirement.

2.2 LBS k-Anonymization

The general framework of LBS k-anonymization systems is shown in Figure 2.2. The framework states a mobile user has a secure connection with a trusted Anonymization Service that acts as a proxy in communicating over an insecure channel with the semi-trusted Location Based Service.

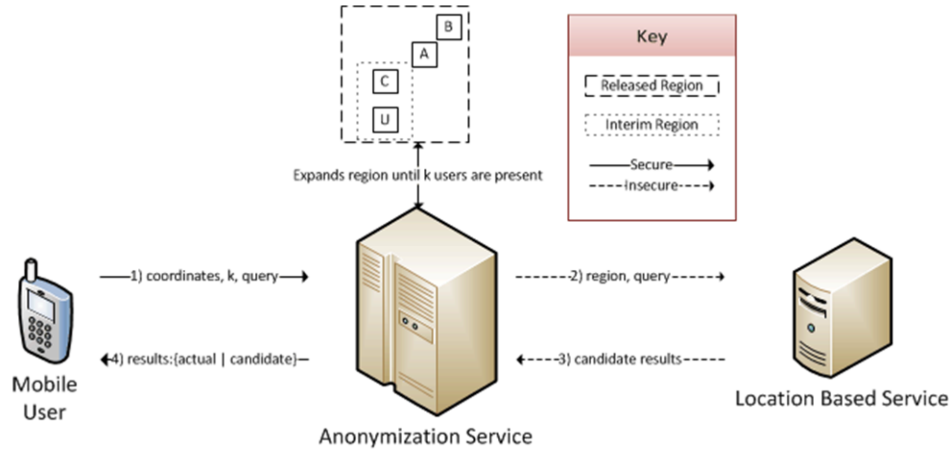

Figure 2.4: General Framework

An example workflow is also shown in Figure 2.2. The Mobile User sends their coordinates, k requirement, and query to the Anonymization Service over a secure connection. The Anonymization Service then anonymizes the Mobile Users coordinates into a region that encompass at least k -1 other users and forwards anonymized query to the LBS provider. The LBS provider processes the anonymized query and returns the candidate results to the Anonymization Service. At this point the candidate results are either filtered at the Anonymization Service and the actual results are sent over the secure connection to the Mobile User, or the full candidate results are forwarded to the Mobile User and the Mobile Users device is responsible for determining the actual result of the query.

2.2.1 Definitions

Definition 2 (Coordinate)**.** Pair of x,y values. Normally latitude and longitude values but this is not required as long as the x and y domains are well-ordered.

Definition 3 (Spatial Region)**.** Spatial area with clearly defined edges, i.e. a coordinate is either inside or outside the region

Definition 4 (Degraded Quality of Service (QoS))**.** Degraded QoS signifies an increase in response time up to no response at all and/or a decrease in accuracy up to completely inaccurate results.

Definition 5 (Maximal Bounding Region (mbr))**.** Total spatial area a Spatial Region may occupy, area is defined in the geometric sense.

Definition 6 (Spatial LBS k-anonymity)**.** Let R be a Spatial Region. Let U be the set of distinct users within R. R is said to satisfy Spatial LBS k-anonymity iff $|U| \geq k$.

Definition 7 (Spatiotemporal LBS k-anonymity)**.** Let R be a Spatial Region. Let t1 and t2 be instances in time s.t. t1t2. Let U be the set of distinct users within R during the interval [t1, t2]. R is said to satisfy Spatial-Temporal LBS k-anonymity $|U| \geq k$.

Definition 8 (Location l-diversity [4])**.** Let R be a Spatial Region. Let L be a set of distinct addressable locations that are within the bounds of R. R is said to satisfy Location l-diversity iff $|L| \geq l$.

Definition 9 (LBS (k,T)-Anonymity [19])**.** Let R be a Spatial Region. Let t be an instant in time s.t. t current time. Let Q be a set of distinct queries released during the interval [t, current time] that have spatial regions which overlap with R. R is said to satisfy LBS (k,T)-Anonymity iff $|Q| \geq k$.

Definition 10 (Extended Spatial-Temporal k-anonymity)**.** Place holder see comments for original definition

Definition 11 (Reciprocity [14])**.** Let R be a Spatial Region. Let U be the set of distinct users contained within the bounds of R. Let I be the user issuing the query. R is said to satisfy Reciprocity iff $I \in U$, $|U| \geq k$, and every member of U modifies their query region to match R.

2.2.2 *Related LBS K-anonymization*

The anonymization algorithm takes a users coordinate as focus and expands into the surrounding spatial area until the users criteria have been met, replace the users identifier with a pseudonym, and replace the original coordinate with the expanded Spatial Region. The region is most often rectangular [4; 13; 19; 11; 10] or circular [14] in shape. Other shapes are not explicitly disallowed however they have not been thoroughly explored. The temporal anonymization is normally a side effect of processing the user's spatial generalization, I.E. waiting on additional users to fulfill the k requirement. Please note that this is contrary to the NLBS approach as a single user's anonymization is the focus, once we are able to anonymize that user we will leave the other user's not in k group as they are until another new user pops into existence at which the process repeats with that new user as the focus. While in NLBS the focus is on anonymizing every user. Please note peer-to-peer based anonymization approaches such as Mobihide [12] exist, however we use a trusted third party anonymizer and therefore will not explore p2p approaches.

Adaptive-Interval Cloaking

Adaptive-Interval Cloaking [13] is based upon Quadtree algorithms and provides both spatial and temporal anonymity based upon the Definition 6 and Definition 7 respectively. Spatial anonymization takes the region surrounding the original query coordinates and sub divides it to the point where the next subdivision would cause the region to no longer satisfy Definition 6. Temporal anonymization takes the region surrounding the original query coordinates and subdivides it to a system specified size at which point the algorithm will hold the query until the region satisfies Definition 7. The k used is a system specified and system wide value. User positions are assumed known at all times to the system.

PrivacyGrid

PrivacyGrid [4; 9] is based upon Grid and utilizes two different approaches, Top-down and Bottom-up algorithms. It provides spatial anonymity based upon the Definition 6, Definition 7, and Definition 8. The Top-down approach begins with the largest possible region permitted by the users mbr and opportunistically erodes the edges of the region until the next erosion iteration would cause the region to no longer satisfy Definition 6. The Bottom-up approach begins with users original cell in the grid and opportunistically expands an edge of the region into the surrounding area until the region satisfies Definition 6. The k, l, and mbr values used are on a per message basis, user positions are assumed known at all times to the system. Regarding the temporal aspect, the algorithm does not go into detail on the use of the allowed delay, and for that reason it is not included in this description.

CliqueCloak

CliqueCloak [11] is based upon clique identification within a graph and provides both spatial and temporal anonymity as defined in Definition 6 and Definition 7. Temporal and Spatial anonymization take place concurrently, the original query coordinate is added into a graph data structure with other queries from distinct users within the mbr are checked to see if they form a clique with original query and if the distinct users k values are less than or equal to the original queries k value. If they do not form a clique, or if the clique size does not satisfy Definition 6, the query is maintained in the graph until its allowed delay value has expired at which time it is removed from the graph. The k, mbr, and allowed delay are on a per message basis. User positions are not assumed known to the system, they are gleaned from the users queries and remain valid until the query is issued or until the allowed delay has expired.

A pitfall with this approach is the formation of the clique, if a message with a large k value arrives prior to a series of messages with smaller k values, the message with the large k value may not be anonymized. For example 1st msg.k=3 and msg.delay=4 arrives at time T=0. Then at T=1 two messages come in with msg.k=2 and msg.delay =2. The messages that arrived at T=1 would anonymized with each other, while the message that came in at T=0 would not be anonymized even though there were enough users present at T=1 to anonymize it.

LBSKT

LBSKT [3] [19] provides both spatial and temporal anonymity as defined by Definition 9. Temporal and Spatial anonymization take place concurrently, the original query coordinates are added into a corresponding cell within a system maintained grid data structure. A region is initialized at this cell and expands until the region satisfies Definition 9. User positions are not assumed known, only their last anonymized querys Spatial Region is known. The k, and T values are designed on a per message basis, however implementation has been done on a system wide k and T value. Another note on implementation is that this system was built upon PrivacyGrid [4].

2.3 Unaddressed in Current Techniques

The approaches of the related works in NLBS and LBS k-Anonymization have identified non-ordered quasi-identifiers such as gender, race, and origin, and ordered quasi-identifiers such as coordinates and time, respectively. However, neither is capable of handling the other's quasi-identifiers elegantly.

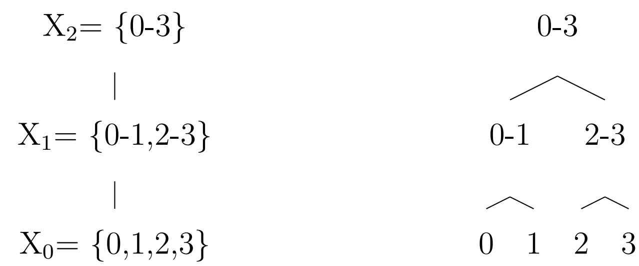

Figure 2.5: Coordinate DGH and VGH

In the case of NLBS k-Anonymity systems tackling the ordered quasi-identifier of coordinates there will be an excessive loss of data accuracy. This loss is due to the tree based generalization mechanism. For example if we use the x component of the coordinates, the DGH and VGH in Figure 2.3, and the data set in Table 2.4 we will generate the k-anonymous X of 0-3. This is an unnecessary full generalization while an LBS centric system would produce an X of 2-3. Less accurate location data leads increased processing time for the LBS provider, anonymization service, and increased data transfer overall.

Name	X	Content
Mike	1	C_m
France	2	C_f

Table 2.4: NBLS Raw User Location Information

The issue of excessive loss of accuracy is not an issue of the chosen DGH or VGH. It results from the tree based structure of the VGH, namely for any VGH on a well ordered set there exists edges between generalization "buckets" and elements falling into said edges will require greater traversal up the VGH. In the above example the edges are 1 and 2.

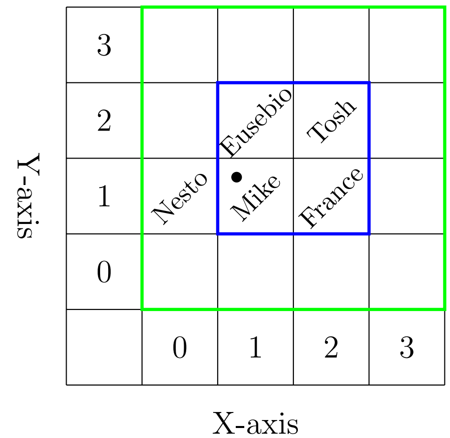

Figure 2.6: User locations

Regarding LBS k-Anonymization of NLBS quasi-identifiers, it is not possible. These approaches take advantage of the ordering of their domains to yield k-anonymous data sets. Determining which of the following integers:{0,1,5} is closer to 2 is simple, however if the domain was race this task becomes impossible. For example with the given information which of the following races:{black,white} is closer to asian. Answering this question is not possible as there is no ordering amongst the elements of the race domain.

If the only quasi-identifier taken into account is the spatiotemporal point of the user as is the case for the current LBS k-anonymization schemes they do not address the possible loss of privacy when additional sensitive attributes are present. Take for example the grid shown in Figure 2.6 that shows the location of the five users presented earlier. The combined data set, location and demographic, is shown in Table 2.5 and the query point is represented by the black dot in Figure 2.6

LBS k-anonymization approaches with k=4 would favor the blue enclosed region shown in Figure 2.6 resulting in the set of queries shown in Table 2.6 to be forwarded to LBS provider. Please note the missing user *Nesto*, the nature of this type of

Name	X	Y	Gender	Race	Origin	Content
Mike	1	1	male	white	United States	C_m
France	2	1	female	black	Haiti	C_f
Eusebio	1	2	male	white	Mexico	C_e
Tosh	2	2	female	native	United States	C_t
Nesto	0	1	female	asian	Mexico	C_n

Table 2.5: Combined Raw User Location Information

anonymization does not necessitate that every element be present in resultant set. It only requires that the initiator user be present in any generated k-anonymous sets. The released table complies with Definition 6 however it does not satisfy Definition 1 , as such the probability of linking the released information to the raw data in Table 2.5 is greater than $\frac{1}{k}$. Please note that row rearrangement has not been preformed to maintain clarity of example.

Name	X_1	X_2	Y_1	Y_2	Gender	Race	Origin	Content
$Pseudonym_{Mike}$	1	2	1	2	male	white	United States	C_m
$Pseudonym_{France}$	1	2	1	2	female	black	Haiti	C_f
$Pseudonym_{Eusebio}$	1	2	1	2	male	white	Mexico	C_e
$Pseudonym_{Tosh}$	1	2	1	2	female	native	United States	C_t

Table 2.6: LBS k-anonymous resultant, k=4

In this situation, due to the extra quasi-identifiers being released, the actual probability is $\frac{1}{1}$ that an attacker can link the user to query. In this attack scenario, the attacker is aware of all the users in this section of the grid, and is knowledgeable about their Gender, Race, Origin. However once the attacker obtains the service request data set, Table 2.6, they are able to link the query with the actual user thanks to

Name	X_1	X_2	Y_1	Y_2	Gender	Race	Origin	Content
$Pseudonym_{Mike}$	0	3	0	3	not_released	not_released	north america	C_m
$Pseudonym_{France}$	0	3	0	3	not_released	not_released	north america	C_f
$Pseudonym_{Eusebio}$	0	3	0	3	not_released	not_released	north america	C_e
$Pseudonym_{Tosh}$	0	3	0	3	not_released	not_released	north america	C_t
$Pseudonym_{Nesto}$	0	3	0	3	not_released	not_released	north america	C_n

Table 2.7: NLBS k-anonymous resultant, k=4

the aforementioned attributes, even though the table has had all direct identifiers removed and the spatiotemporal region has been anonymized. Essentially the attacker is able to map the pseudonym'd user to actual user.

NLBS k-anonymization approaches with k=4, the coordinate generalization scheme in Figure 2.3, and the each columns respective scheme presented earlier in this chapter would favor the green enclosed region shown in Figure 2.6 resulting in the set of queries shown in Table 2.7 to be forwarded to LBS provider. The released table complies with both definitions 6 and 1 , as such the probability of linking the released information to the raw data in Table 2.5 is $\frac{1}{k}$. Please note that row rearrangement has not been preformed to maintain clarity of example.

The extra location quasi-identifier did not affect the probability of a linking attack as occurred in the LBS case. However, the extra location information is off little to no use. It is fully generalized, increasing processing time throughout the anonymization service and LBS provider. At this level of accuracy the presence of location data is a moot point at best and a waste of processing power at worst.

Chapter 3

FRAMEWORK FOR K-ANONYMOUS M-COMMERCE

The approach taken marries NLBS k-Anonymity and LBS k-Anonymity in the LBS environment that meets or exceeds the cloaked users' anonymization policies. This marriage allows for handling of both location and demographic quasi-identifiers. The necessity to include more quasi-identifier types than the current LBS k-anonymization approaches results from the nature of M-Commerce, Targeted Advertising, and future context sensitive LBS providers. In these domains the service providers require more information than location in order to function. The extra required quasi-identifiers such as gender, age, and ethnicity may used in a linking attack to erode the privacy of a user.

As discussed previously each anonymization domain handles different types of quasi-identifiers. NLBS uses a tree structure for generalization that is geared for quasi-identifier domains with no ordering. While LBS leverages the well-ordered nature of location information for generalization. Essentially NLBS will always follow the same generalization path for a given value while LBS can yield many different generalization paths for a given value. In addition to this the NLBS k-anonymization normally is solving the problem of finding a single generalization scheme that satisfies the k requirement for every member of a data set. LBS on the other hand locates a generalization scheme that satisfies k for a subset of the original data set at a time.

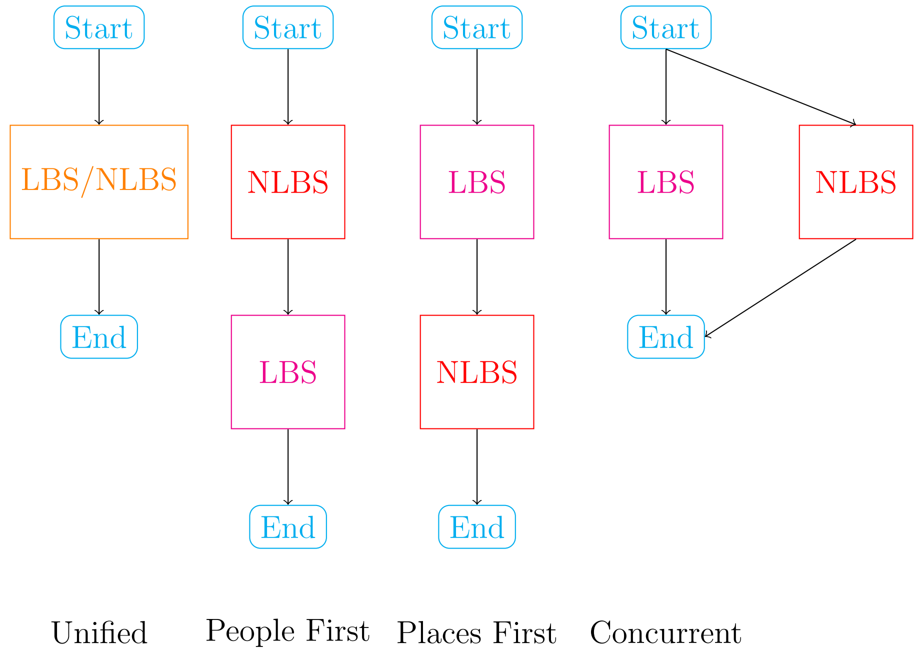

Figure 3.1: Combination Possibilities

3.1 Anonymization Service

Minimal generalization is the primary concern of the anonymization service. Rapid processing is a close secondary concern. As the data becomes less accurate the LBS provider is burdened and the amount of post filtering of the LBS providers resultant is increased. The rate of anonymization directly affects the types of services that end-users can utilize. Targeted advertising in M-commerce is highly dependent upon the end-user's current environment. Small changes in location, occurring in a manner of seconds, may greatly affect the ad content.

3.1.1 Possible NLBS/LBS k-Anonymization Combinations

Due to the various types of quasi-identifier domains and existence of approaches to handle each class of domain there are four combinations possible as shown in Figure 3.1. We will describe each in the following sections.

Unified

The first combination merges NLBS and LBS k-anonymization concepts into a single algorithm. The initial data set is transformed into an k-anonymous one by a single unified approach. Further discussion located in Algorithm and Data Structures.

People First

In this combination the data set is first sent through an NLBS based k-anonymization approach then those results are fed to an LBS based k-anonymization scheme to produce a k-anonymous resultant. However, the intermediate table produced by the NLBS approach is not closed under subset. Without closure under subset there is no guarantee the LBS k-anonymization resultant will still be k-anonymous with respect to the NLBS quasi-identifiers. As the NLBS k-anonymization will not take location into account, it will produce an anonymous data set containing every user present regardless of their proximity to each other. When this data set is processed by the LBS k-anonymization approach k users in the resultant will be selected to minimize the spatial area needed to encompass them. The k users selected are a subset of the NLBS k-anonymous data set. We will prove the non-closure by assuming closure and providing a counter example where the above described process produces a data set that is not k-anonymous.

Take the data set from Table 2.5 and produce a NLBS k-anonymous data set with k=2 and the VGHs from Figures 2.1.1, 2.1.1, and 2.1.1. Please see Table 3.1 for the resulting data set. Now assuming our focus is *Nesto* when we produce the LBS k-anonymous data set we will group *Mike* with *Nesto* as he is closer than any other user. As can be seen in table the resultant is LBS k-anonymous but it is not NLBS k-anonymous as it does not satisfy Definition 1. Namely, they are distinguished by gender.

Name	X	Y	Gender	Race	Origin	Content
Mike	1	1	male	not_released	north america	C_m
France	2	1	female	not_released	north america	C_f
Eusebio	1	2	male	not_released	north america	C_e
Tosh	2	2	female	not_released	north america	C_t
Nesto	0	1	female	not_released	north america	C_n

Table 3.1: NLBS k-Anonymized k=2

Name	X_1	X_2	Y_1	Y_2	Gender	Race	Origin	Content
Mike	0	1	1	1	male	not_released	north america	C_m
Nesto	0	1	1	1	female	not_released	north america	C_n

Table 3.2: LBS k-Anonymized k=2 Unsuccessful

Places First

Contrasting *People First*; An LBS k-anonymous data set is generated then input to an LBS k-anonymization algorithm to produce an LBS/NLBS k-anonymous resultant. This approach was pursued initially but due to poor performance exploration was discontinued.

Separate But Equal

Finally in this combination LBS and NLBS approaches are run in parallel and their results are then joined in the relational algebra sense to produce a k-anonymous data set. Exploration of this approach has been slated for future work.

3.1.2 Algorithm and Data Structures

Among the aforementioned combinations we focus on *Unified* for the reasons presented in the preceding section. As previously stated this is a novel approach unifying the concepts from both LBS and NLBS k-anonymization. Like LBS it operates with a single user focus resulting in a set of anonymized messages containing the original user and at least k-1 other users. The resultant is constructed by gathering a group of users message's with the minimal amount of *difference*. The group's generalization is applied by looking at the overall messages quasi-identifier extremities and locating a suitable generalization with regards to those two values.

Definition 12 (Difference)**.** Measurement of generalization required to make entities indistinguishable by their quasi-identifiers.

The pseudo-code for this approach is located in Algorithm 1. It is composed of several sub-algorithms, presented in the order they appear in Algorithm 1, and operates on messages structured as below. Where M is set of incoming messages to the anonymization service; m_M is a message in M; ID is a unique identifier; $\{x, y\}$ is the originating point of the message; QI_x is a quasi-identifier; t is the allowed amount of delay; k is required number of indistinguishable users for any resultant containing this message; C is content of message.

$$m_M \in M : \langle ID, \{x, y\}, \{QI_0, QI_1, ..., QI_{n-1}\}, t, k, C\rangle$$

The algorithm maintains Q_m and $UNANONYMIZED$. Q_m is a queue of incoming messages that have yet to be processed. As a queue it is FIFO with the ordering based upon arrival time to the anonymization service. $UNANONYMIZED$ is a fully connected graph structure containing messages which have been unsuccessfully processed. The edges represent the difference between the messages they connect.

The algorithm first checks for a message in Q_m then selects the oldest message for processing, lines 1 and 2 respectively. Line 4 ensures the message originated from a location managed this anonymization service. Line 5 generates an interim fully connected graph structure G_t that is a clone of $UNANONYMIZED$ with the message included. Line 6 is an optimization that prevents processing a message if it's k requirement is unsatisfiable. In this event the message is added to $UNANONYMIZED$. Lines 6 through 18 repeatedly generate a candidate set of messages from G_t, unify the k policy of generated set, and remove messages with stricter k requirements than the generated set can satisfy from G_t until either a suitable set is found or finding one becomes impossible. A suitable set is found when the generated set's cardinality is $\geq$ the unified k while impossibility condition is triggered when the number of messages in G_t falls below the unified k requirement. Line 19 branches on the success of Lines 6 through 18. If it was unsuccessful the message is incorporated into $UNANONYMIZED$ then returns null. If it was successfully the candidate set along with the message are generalized then returned.

Calculating the difference between messages is performed by the *Calculate Difference* Algorithm 2. It is essentially a weighted euclidean distance formula, where each quasi-identifier in has an associated weight $\omega(D_i)$ and difference function $d(p_i, q_i, D_i)$. Please see Algorithm 3 and Algorithm 4 for the weight and difference function respectively.

The weights given to a quasi-identifier domain depend on type and represent the inverse of the maximal change required for any two values to become indistinguishable.

Algorithm 1 LIVE

if $Q_m \neq \emptyset$ **then**
  $m_0 \leftarrow$ Pop first message of Q_m
  $m_0 \leftarrow$ Append r,c to m_0 where $r = row(m_0.y)$ and $c = column(m_0.x)$
  $returnable \leftarrow \emptyset$
  if $m_0.coordinates \in REGION$ **then**
    Add m_0 into G_w
    for all $m_i \in UNANONYMIZED$ **do**
      if $(m_0.mbr \geq |m_0.c - m_i.c|) \& (m_0.mbr \geq |m_0.r - m_i.r|)$ **then**
        if $(m_i.mbr \geq |m_0.c - m_i.c|)$ & $(m_i.mbr \geq |m_0.r - m_i.r|)$ **then**
          $d \leftarrow calculate\_difference(m_0, m_i)$
          Add edge (m_0, m_i) with weight d to G_w
    Add m into UNANONYMIZED
    if $G_w(m_0).edges >= (m_0.k - 1)$ **then**
      $clique \leftarrow get\_k\_clique(G_w, m_0)$
      if $clique \neq \emptyset$ **then**
        $returnable \leftarrow apply\_generalization(clique, k)$
        for all $m_i \in clique$ **do**
          Remove m_i from G_w
          Remove m_i from UNANONYMIZED
  return $returnable$

Algorithm 2 Calculate Difference

$$calculate\_difference(p, q) = \sqrt{\sum_{i=1}^{n} \omega(D_i)^2 d(p_i, q_i, D_i)^2}$$

In other words the percentage of accuracy lost for each generalization increase.

As shown in Algorithm 3 a finite well-ordered set such as latitude or age uses $\frac{1}{|D|-1}$. The cardinality of the domain represents the total number of elements that can included while generalizing. We remove one from the total to compensate for the initial value. The inverse is used as it is the percentage of generalization or inaccuracy that each additional element adds when included in the result.

Algorithm 3 Weight

$$\omega(D) = \begin{cases} \frac{1}{|D|-1} & \text{if D is finite well-ordered set} \\ \frac{1}{ht(D)} & \text{if } |D| > 1 \text{ \& D is finite hierarchical (partially ordered) set} \\ 1 & \text{if D is finite unordered set} \end{cases}$$

The finite hierarchical set shown in Algorithm 3 has a total number of generalization steps equal to the height of the VGH or DGH. Illustrating this is the Gender domain shown in Figure 2.1.1, there is only a single generalization step for the leaves male and female in the VGH and there is only a single step in the DGH. This results from the from definition of Domain Hierarchies in [8].

Contrary to the hierarchical and well-ordered sets the size of the unordered domain has no bearing on the generalization level. In this case the amount of information lost is completely dependent upon the two users being compared as a user is unable to lose information that they never had in the first place. The unordered set views the percentage of inaccuracy increased per element removed. As it is removal being measured the initial values may be removed, hence no need to deduct one. While this type of domain has not been explored as of yet it does extend the expression capabilities of k-anonymization. Take for example the multiracial individuals shown in Table 3.3 and the race hierarchies in Figure 2.1.1. Currently the leaves of the VGH are single elements and not sets, therefore there is no specified manner to generalize in these situations. An approach to accommodate these data types is presented later this section.

Name	Gender	Race	Origin	Content
Mike	male	{white}	United States	C_m
France	female	{black,white}	Haiti	C_f
Eusebio	male	{white}	Mexico	C_e
Tosh	female	{native,white,asian}	United States	C_t
Nesto	female	{white,asian}	Mexico	C_n

Table 3.3: Multiracial Modified NBLS Raw User Information

The Difference algorithm 4 determines the number of generalization steps required to make a pair of quasi-identifier elements from the same domain indistinguishable. The inner working of the algorithm are dependent upon the type of domain being operated on.

Algorithm 4 Difference

$$d(p,q,D) = \begin{cases} index(p) - index(q) & \text{if D is finite well-ordered set \& p,q non-set elements} \\ ht(LCA(p,q)) & \text{if } |D| > 1 \text{ \& D is finite hierarchical} \\ & \text{(partially ordered) set \& p,q non-set elements} \\ 1 - \frac{|p \cap q|}{|p \cup q|} & \text{if D is finite unordered set} \end{cases}$$

In the case of finite well-ordered sets we take the difference of the index or ordinal value of p and q. As the set is well-ordered the amount of generalization required to make p and q indistinguishable is equal to the number of items between p and q. Conceptually we are increasing the interval size until enough users fall within it's boundaries. For example in Figure 2.6 the domain for the quasi-identifier X is the set {0,1,2,3}. *Nesto*, *Eusebio*, and *France* are located at 0, 1, and 2 respectively. The difference between *Nesto* and *Eusebio* is 1 while between *Nesto* and *France* it is 2. If we were to want a $k = 2$ with respect to the X axis and *Nesto* as the initiator an anonymized value of *0-1* or *0-2* would satisfy the requirement as both have at least two users whose x point fall within the boundaries of the anonymized interval.

As described in [8] the minimal number of generalizations needed to be preformed on a pair of leaves p and q in a VGH is the height of the lowest common ancestor of p and q. For example in Figure 2.1.1 the $LCA(antigua\&barbuda, unitedstates)$ is *north america* with a height of one. Hence *antiqua&barbuda* and the *united states* become indistinguishable at one step of generalization.

The unordered set works in an opposite conceptual manner to both hierarchical and well-ordered sets. The other types view generalization as the inclusion of extra elements in the anonymized resultant. Well-ordered will grow the interval size while traversal up a tree in the hierarchal set will increase the number of leaves. Unordered sets on the other hand will remove elements from their initial sets in order to produce the k-anonymous result. The intersection of the initial sets gives the overlapping values which can be release without distinguishing either original set. For example using Table 3.3 as a source set, $k = 2$, ignoring direct identifiers, and the quasi-identifier of race we can produce the NLBS k-anonymous Table 3.4. This table shows more racial information than that of Table 2.2.

Name	Gender	Race	Origin	Content
Mike	male	{white}	north america	C_m
France	female	{white}	north america	C_f
Eusebio	male	{white}	north america	C_e
Tosh	female	{white}	north america	C_t
Nesto	female	{white}	north america	C_n

Table 3.4: Multiracial Modified NBLS k=2

The amount of information lost is equal to one minus the percentage of remaining diversity of the resultant set. The quantity $\frac{|p \cap q|}{|p \cup q|}$ shows the remaining diversity percentage. $|p \cap q|$ is a count of the releasable elements while $|p \cup q|$ is the maximum number of elements available for release. The amount of information loss in unordered domains is respective to the sets being compared and not to the domain itself.

The actual search for the k clique is performed by algorithm an A* based approach. As A* is geared for path finding in graphs it works on the concept of a source and goal vertex. Though it appears that we have all the components save for a terminal or goal vertex we do not. The former we have as m however we do not have a goal vertex. Therefore we defined goal vertex conditions, namely we know we have arrived at a goal vertex when we have reached our required k size. With both the source vertex and terminating conditions present the algorithm then searches through the possible set of members of the k clique by adding a single member at a time to the initial set containing m.

The additional members loosely come from the original graph G, however the A* algorithm is not directly searching G it is in fact exploring many derivatives of G. At any given step in the process the next possible member to the clique is dependent both upon existence of arcs between all members of the current clique elements and the possible member as well as the policy constraints on the resulting clique.

An initial state is created in lines 5-11. The states maintain a spatial region defined by lower and upper rows as well as columns of the messages present in the state, this is initialized in line 5. Policy values of mbr and k are also set in line 5. As this is an A* algorithm the associated g, h, f values are set in lines 6, 7, and 11 respectively. Each state is also responsible for maintaining the members present in the k clique for the state, as each additional member will signify a new state. The members present set is initialized in line 8 to that of the source message m. The state is added to the *openlist* structure of the A* in line 12 and as we do not have a goal state but rather terminal conditions we will repeatedly loop through lines 14 - 44 until found which is set to false in line 13 is set to true by the terminal conditions being met. Line 9 sets the handle for the vertex representing the set of nodes in the candidate clique in the states associated G.

The inner workings of the loop is an extension the A* algorithm. The lowest F valued state is popped from the *openlist*, line 15. Then it is checked to see if it satisfies the terminal conditions in line 16. The state's h being 0 signifies that it is already satisfied with the members present. If it meets the terminal conditions then found is set to true, the nodes present in the clique managed by the state are returned and then removed from the original graph G; lines 17 through 19 respectively. If not then additional states are generated and checked.

Algorithm get_k_clique: part 1

```
function GET_K_CLIQUE(G, m)                                ▷ A* search based approach
    returnable ← ∅
    openlist ← Empty priority queue
    closedlist ← ∅
    initial_state(state_0, m)
    state_0.g ← 0
    state_0.h ← H(G, m, state_0.k − 1)
    state_0.nodes ← {m}
    state_0.label ← m
    state_0.G ← G
    state_0.f ← state_0.g + state_0.h
    Push state_0 to openlist
    found ← false
    while !found do
        state_c ← openlist.pop()
        if state_c.h == 0 & |state_c.nodes| ≥ state_c.k then
            found ← true
            returnable ← state_c.nodes
            Remove all state_c.nodes from G
        else
            Add state_c to closedlist
            for all neighbor w of state_c.label ∈ state_c.G do
                state_n.nodes ← state_c.nodes ∪ w
```

Algorithm get_k_clique: part 2

$valid \leftarrow enforcers\_statisfied(state_c, state_n, w)$

if $!(state_n \in closedlist)$ & $valid$ **then**

  $state_n.G \leftarrow getMergedCopy(state_c.G, state_c.label, w)$

  $state_n.label \leftarrow min(state_c.label, w)$

  $state_n.h \leftarrow H(state_n.G, state_n.label, state_n.k - |state_n.nodes|)$

  $state_n.g \leftarrow state_c.g + weight(Edge(state_c.label, w) \in state_c.G)$

  $state_n.f \leftarrow state_n.g + state_n.h$

  if $!(state_n.h < 0)$ **then**

    if $state_n \in openlist$ **then**

      if $state_n.g < openlist[state_n].g$ **then**

        Remove $openlist[state_n]$ from $openlist$

        $state_n.parent \leftarrow state_c$

        Add $state_n$ to $openlist$

    else

      $state_n.parent \leftarrow state_c$

      Add $state_n$ to $openlist$

  else

    Add $state_n$ to $closedlist$

else

  Add $state_n$ to $closedlist$

$found \leftarrow isEmpty(openlist)$

return $returnable$

end function

As this state does not satisfy the terminating conditions and it has been checked already it is now added to the closed list; line 21. The additional states are generated by looking to label/vertex and graph of this state and enumerating the neighbors of the vertex in the graph; line 22. Each enumerated neighbor will first produce a state shell that is used for comparison and policy enforcement; lines 23 trough 24. The new state's clique is the union of the preceding state's clique and the enumerated neighbor, line 23. As all messages have associated row and column data the spatial region of the new state is updated in line 24. The new state's k and mbr policies are updated, checked against the new state, and checks resultant returned in line 24.

Line 25 ensures we are not rechecking a visited state and that new state is valid, if not then the state is added to the *closedlist* else the new state is promoted from a shell to a full state; lines 26 through 30. This entails generating a $graph, label, h, g$, and f for the newly promoted state. Line 31 ensures only possibly satisfiable states are added to the *openlist*. If the state is unsatisfiable then it is added to the *closedlist*; line 41. The lines 32 through 41 update the *openlist*, by adding a new state, lines 38 - 39, or by replacing a state currently present, lines 34 - 36.

As get_k_clique algorithm is based upon A* we a need a heuristic or future cost function (h), past path cost function (g), and the composite cost function (f). f is the sum of the cost to our current state and the estimated cost from said point to the goal state thus the equation $f = g + h$. The past path cost is initialized at 0 as when we start we have yet to travel, and sums the edge weights present on the path selected. $g_n = g_{n-1} + weight(Edge(g_{n-1}.label, x) \in G_{n-1})$ with $g_0 = 0$ where G_{n-1} is the graph associated with $state_{n-1}$.

Algorithm initial_state

1: **function** INITIAL_STATE(state, m)

2: $state.row_{lower} \leftarrow m.row$

3: $state.row_{upper} \leftarrow m.row$

4: $state.col_{lower} \leftarrow m.col$

5: $state.col_{upper} \leftarrow m.col$

6: **if** $(m.mbr == \emptyset)|(m.mbr < 0)$ **then**

7: $\quad m.mbr \leftarrow \infty$

8: $state.mbr \leftarrow m.mbr$

9: **if** $(w.k == \emptyset)|(w.k < 0)$ **then**

10: $\quad w.k \leftarrow 0$

11: $state_0.k \leftarrow m.k$

12: **end function**

The *initial_state* Algorithm initializes the state's policy values based upon the message m, lines 8 and 11. A sanity check is preformed on the mbr, lines 6-7, and k, lines 9-10, policies. As the mbr is a constraint on the spatial region area we create a boundary box around m using m's row to set bottom and top edges, lines 2 and 3 respectively. Equivalently we set left and right edges using m's column in lines 4-5.

The heuristic function presented in Algorithm 3 under estimates the cost from the vertex in a given graph to the goal state. It sums the edge weights between the vertex and the number of other vertices needed to reach the required k value as having a group of size k is the goal state. The summation is done in ascending order of edge weights, so the minimum weighted edges are included first as in an ideal situation these would be the vertices selected for the group due to the minimal *difference* from our current group. In addition to the summation the difference between the final or

Algorithm 3 Heuristic

function H(G,vertex, required)

  $returnable \leftarrow 0$

  $nearest \leftarrow$ List sorted in ascending weights order of neighbors of $vertex \in G$

  if $|nearest| \geq required \ \& \ |nearest| > 0$ **then**

    for all $n = 0; n < required; n++$ **do**

      $returnable += weight(Edge(vertex, nearest[n]))$

    $returnable += weight(Edge(vertex, nearest[required - 1])) - weight(Edge(vertex, nearest[0]))$

  else

    $returnable \leftarrow -1$

  return returnable

end function

max edge weight and the initial or min edge weight is also included as we can infer via the triangle inequality that between these two vertices there is an edge of at least this length. In the event that the vertex does not have enough neighbors then -1 is returned as a sentential value signifying that there is not a route from the vertex to a satisfied state.

The enforcers_satisfied function sets and checks the next state's policies. The *mbr* policy requires the row and column values for the new state to be set, lines 2-5. If a policy field isn't present or value is below zero, the policy is set it most lenient settings for the user, lines 6-9. A value of infinite for *mbr* allows the full spatial region to be explored; k of 0 requires no anonymity. The new state's k policy should be the maximum of preceding state's k and enumerated neighbors k, line 10. While the *mbr* policy should be the minimum of the preceding state's *mbr* and enumerated

Algorithm enforcers_Satisfied

1: **function** ENFORCERS_SATISFIED($state_c$,$state_n$, w)

2: $\quad state_n.row_{lower} \leftarrow min(state_c.row_{lower}, w.row)$

3: $\quad state_n.row_{upper} \leftarrow max(state_c.row_{upper}, w.row)$

4: $\quad state_n.col_{lower} \leftarrow min(state_c.col_{lower}, w.col)$

5: $\quad state_n.col_{upper} \leftarrow max(state_c.col_{upper}, w.col)$

6: $\quad$ **if** $(w.mbr == \emptyset)|(w.mbr < 0)$ **then**

7: $\qquad w.mbr \leftarrow \infty$

8: $\quad$ **if** $(w.k == \emptyset)|(w.k < 0)$ **then**

9: $\qquad w.k \leftarrow 0$

10: $\quad state_n.k \leftarrow max(state_c.k, w.k)$

11: $\quad state_n.mbr \leftarrow min(state_c.mbr, w.mbr)$

12: $\quad valid \leftarrow state_n.row_{upper} - state_n.row_{lower} \leq state_n.mbr$

13: $\quad valid \leftarrow (state_n.col_{upper} - state_n.col_{lower} \leq state_n.mbr)\ \&\ valid$

14: $\quad$ **return** $valid$

15: **end function**

neighbors mbr, line 11. The mbr constraint is now checked in lines 12 through 13. If the difference between the upper and lower rows of the new states spatial region is less than or equal the new state's mbr and the same holds true for the columns then this new state is a valid state.

Algorithm 3 creates a new graph derived from the G with $vertex1$ and $vertex2$ merged into a single vertex. The set edges of the new vertex are the intersection of the sets of $vertex1$ neighbors and $vertex2$ neighbors. Edge weight is sum of weights from corresponding edge in $vertex1$ neighbors and $vertex2$ neighbors. Line 4 ensures that the created vertex has a predictable name for access later. Non-mutual adjacent

Algorithm 3 get_Merged_Copy

function GET_MERGED_COPY(G,vertex1,vertex2)

    $G_r \leftarrow$ copy of G

    Remove $vertex1, vertex2$, and all associated edges from G_r

    Create $min(vertex1, vertex2)$ in G_r

    $mutual \leftarrow$ (neighbors of $vertex1 \in G$) $\cap$ (neighbors of $vertex2 \in G$)

    $mutual \leftarrow mutual - \{vertex1, vertex2\}$

    for all $v \in mutual$ **do**

        $w \leftarrow weight(Edge(vertex1, v) \in G) + weight(Edge(vertex2, v) \in G)$

        Add edge $(min(vertex1, vertex2), v)$ with weight w in G_r

    return G_r

end function

vertices are excluded as neighbors of the merged vertex because their inclusion would violate the *mbr* policy of the non-mutual neighbor and that of $vertex1$ or $vertex2$.

Once the k-clique is found it is generalized via Algorithm which returns an anonymized set mapped to their original messages. First an empty set V_i is created for each domain D_i, lines 3-4. Messages are then iterated through and each of their domain values are added to their corresponding set initialized above, lines 5-7. Next each set V_i and its corresponding domain D_i are fed to *get_generalization* which produces a minimal anonymized value L_i for each D_i given its associated V_i, lines 8-9. Then the mapped table is constructed via iterating through the original messages and deriving the anonymized version, lines 10-15. A new message m_c is created, it's identifiers set to a suppressed version of the original message m_s, it's k is set, each domain value $m_c.D_i$ it's respective anonymized version L_i, and finally the new message is mapped to the original, lines 11-15 respectively.

Algorithm apply_generalization

```
1: function APPLY_GENERALIZATION(C,k)
2:     returnable ← deep copy of C
3:     for all D_i ∈ D do
4:         V_i ← ∅
5:     for all m_c ∈ C do
6:         for all D_i ∈ D do
7:             V_i ← m_c.D_i ∪ V_i
8:     for all D_i ∈ D do
9:         L_i ← get_generalization(V_i, D_i)
10:    for all m_s ∈ returnable do
11:        m_c ← Suppress direct identifiers of m_s
12:        m_c.k ← k
13:        for all D_i ∈ D do
14:            m_c.D_i ← L_i
15:        returnable ← returnable ∪ m_c
       return returnable
16: end function
```

The sub algorithm takes in a set of values V and a domain D then produces a single value minimal generalization. As with other algorithms dependent upon the domain it is essentially a piecewise function. Lines 3-12 handle well-ordered domains such as *latitude*; Lines 13-20 operate on partially-ordered domains such as *origin*. Currently unordered domains have not been explored, but intuitively it would be the intersection of values in V.

Algorithm get_generalization

function GET_GENERALIZATION(V,D)

  $returnable \leftarrow \emptyset$

  if D is well-ordered set **then** $e_{min} \leftarrow V[0]$ $e_{max} \leftarrow V[0]$

    for all $e \in V$ **do**

      $e_{min} \leftarrow min(ordinal(e_{min}, D), ordinal(e, D))$

      $e_{max} \leftarrow max(ordinal(e_{max}, D), ordinal(e, D))$

    if $ordinal(e_{min}, D) \neq ordinal(e_{max}, D)$ **then**

      $min \leftarrow string(e_{min})$

      $max \leftarrow string(e_{max})$

      $returnable \leftarrow min + string(-) + max$

    else

      $returnable \leftarrow string(e_{max})$

  if D is partially-ordered set **then**

    $returnable \leftarrow \emptyset$

    for all $e_0 \in V$ **do**

      for all $e_1 \in V$ **do**

        if $returnable == \emptyset$ **then**

          $returnable \leftarrow LCA(e_0, e_1, D)$

        $returnable \leftarrow LCA(LCA(e_0, e_1, D), returnable)$

  return $returnable$

end function

As the well-ordered domains can contain elements of any type, we look at the ordinal position of each value in V to determine the minimum and maximum values, read lowest and highest ordinal, lines 4-6. If the values are different then the generalized resultant should be the interval $[minimum - maximum]$, else it is merely $maximum$. Please note that interval is inclusive and the choice of $maximum$ over $minimum$ for the else section is arbitrary.

The hierarchal or partially ordered domains are generalized by comparing every value in V to every other value in V and determining the their overall lowest common ancestor (LCA). This is a brute force approach, two values e_0, e_1 are chosen from V, $LCA(e_0, e_1)$ is calculated and stored LCA_p, line 18. Then two more values e_0, e_1 are chosen from V and $LCA_p \leftarrow LCA(LCA(e_0, e_1), LCA_p)$, repeat until e_0 and e_1 have assumed all values in V.

3.2 End-user

Privacy erosion, power consumption, and network data costs are concerns of the end-user. The more information a user yields in exchange for a service the greater the loss of privacy. As the end-user operates a mobile device power consumption of processing large result sets from an LBS provider is detrimental to service adoption. In addition to the power consumption, transferring large results sets incurs carrier charges in most cases.

Each LBS provider maintains a unique policy of manging user data. Policies are malleable to business interests and not user-centric. Monitoring the various terms and conditions an end-user's data is subject to for changes is a difficult task. Specifying their own policy and anonymizing their data reduces the privacy risk incurred by an end user.

Battery life and data usage are concerns for the typical mobile end-user. Processing and data usage increase with the size of the LBS anonymous query resultant. The size of the query resultant increases as the generalization of query parameters increases. Filtering the resultant at the anonymization service reduces both processing and data usage at the end-user.

3.3 LBS Provider

LBS providers feature varied services dependent upon the user's location. Detailing their inner workings is difficult due to the diversity of services offered and their associated operating domains. In general LBS providers take a user's GPS coordinate, process it, and yield a result. The processing may range from querying a database, Location Based Access Control enforcement. The results of processing may be returned to the user, passed to another service, and/or simply stored.

In order for these systems to work with our or any Anonymization Service they must be able to process anonymous queries. In a nut shell, instead of receiving exact latitude longitude coordinates the LBS provider will receive a spatial region containing the user's location. The greater the area given to the provider the larger the result and computer cycles consumed.

Thanks to the advertisement driven business model pushed forward by Google and Android, user information is monetizable. As stated previously the LBS provider may send the query containing user information to a third party service such as advertisers, over which the end user has no control of data usage, save for those protections offered by impermanent terms and conditions of the LBS provider. Therefore the LBS provider is considered Semi-Trusted.

Chapter 4

IMPLEMENTATION

We have implemented our proof of concept Anonymization Service in Java. In order to support the varied quasi-identifier domain types the Anonymization Service is comprised of three packages: *Anonymizer*, *Domains*, and *Enforcers*. Please see Figure 4.1 for the dependencies amongst the components. Currently the service supports both NLBS and LBS quasi-identifier domain types, namely: *age*, *gender*, *origin*, *race*, *latitude*, and *longitude*. In addition to this the policy values of k and mbr are also supported. The components are discussed in further detail below.

The *Anonymizer* uses Jgrapht [1] for pending message graph manipulation. As messages arrive their difference to all other pending is calculated using the *Domains* module. While searching for a suitable generalization group the *Enforcers* component is used to determine next state validity and satisfaction. Once a group has been discovered and the anonymized data determined the Anonymizer produces the anonymous queries, maintains a mapping for to the original queries, and removes the group from the pending messages graph. In addition, messages outside the spatial region of the Anonymizer are filtered out prior to processing.

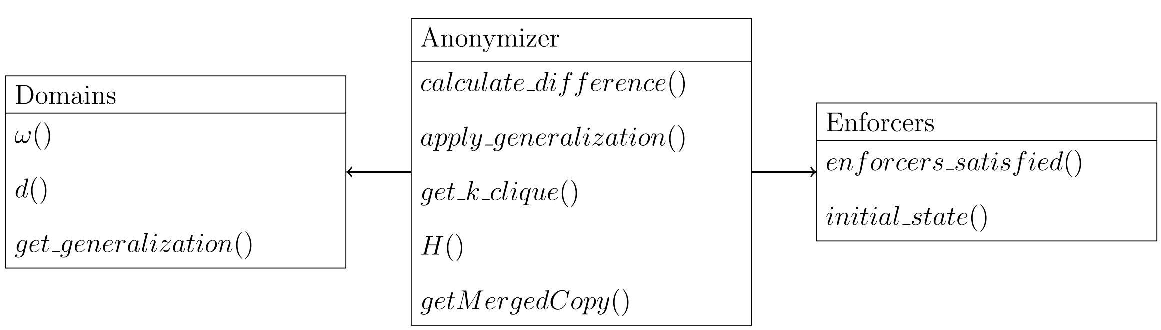

Figure 4.1: Package Diagram

The *Domains* handle the quasi-identifiers and use a different underlying data structure for each domain type. Partially ordered NLBS domains are represented by a hierarchy backed by Jgrapht [1]. Well ordered LBS domains are Lists. The unordered domain data would be managed by a SET type structure. This component generates the difference measure for NLBS domains by looking for the LCA of the two corresponding message components. LBS by getting the ordinal value difference between the corresponding components. Unordered domains by comparing the intersection and union of corresponding component. The weights are a function of the number of possible generalization any initial value may yield. Please see algorithms 2 and 3 for further details.

The *Enforcers* manage the k and *mbr* policies. k is understood to be required size of anonymized group while *mbr* is a cap on the spatial area encompassing the group. As new members are incorporated the groups policy values adjust such that the combined group policy will satisfy all individual member policies. Intuitively selecting the largest k and smallest *mbr* of the group accomplishes this goal.

As this implementation is focused on the Anonymization Service extension we have a wrapper application that reads in user queries from a file and then calls the Anonymization Service to process. The Anonymizer itself is headless and returns the generalized queries mapped to the original to the wrapper application which in turn writes them to a file. As is show in Figure 4 the queries are structured as name, age, gender, race, origin, latitude, longitude, row, col, k, mbr, pseudonym, arrival time, anon start, wait time. The fields of row, col, pseudonym, arrival time, anon time, and wait time are added by the Anonymization Service and are not present in the original query from the user. The Anon start field shows when the message began anonymization processing.

NAME	AGE	GENDER	RACE	Origin	Latitude	Longitude	ROW	COL	K	MBR	PSEUDONYM	ARRIVAL TIME	ANON START	WAIT TIME
Audrey TURNER	33	female	white	Canada	37.578832	-121.752963	168	188	3	35	cd9fd3aa-19c1-4daf-a658-3dd1a158d2c7	1394665675896	1394665675896	0
Alex STEWART	20	male	white	Greece	37.585012	-121.959282	170	157	1	36	537175f5-33ec-4d8f-ada0-c4b7393959f9	1394665675865	1394665675896	31
Morgan ALEXANDER	22	female	white	Greece	37.654132	-121.9871709999	188	153	3	35	91e078be-f490-472c-8787-8f7f104e5509	1394665675880	1394665675896	16

Figure 4.2: Incoming Queries

NAME	AGE	GENDER	RACE	Origin	Latitude	Longitude
cd9fd3aa-19c1-4daf-a658-3dd1a158d2c7	20-33	n_r	white	world	37.57638770949721-37.65726117318436	-121.99015333333332--121.75225999999999
537175f5-33ec-4d8f-ada0-c4b7393959f9	20-33	n_r	white	world	37.57638770949721-37.65726117318436	-121.99015333333332--121.75225999999999
91e078be-f490-472c-8787-8f7f104e5509	20-33	n_r	white	world	37.57638770949721-37.65726117318436	-121.99015333333332--121.75225999999999

Figure 4.3: Outgoing Queries: part 1

ROW	COL	K	MBR	PSEUDONYM	ANON START	ANON FINISH	ANON TIME
-1	-1	3	-1	cd9fd3aa-19c1-4daf-a658-3dd1a158d2c7:ALIAS	1394665675896	1394665675913	17
-1	-1	3	-1	537175f5-33ec-4d8f-ada0-c4b7393959f9:ALIAS	1394665675896	1394665675913	17
-1	-1	3	-1	91e078be-f490-472c-8787-8f7f104e5509:ALIAS	1394665675896	1394665675913	17

Figure 4.4: Outgoing Queries: part 2

The generalized queries presented in Figures 4.3 and 4.4 follow the same message structure as the originals but with different values representing the group instead of the individual with the exception of name and pseudonym. The name receives the originals pseudonym and the pseudonym value get ":ALIAS" appended to it signifying it is the anonymized version of the original message with the pseudonym matching that which immediately precedes ":ALIAS". The fields released to an LBS provider from the anonymized query are only name, age, gender, origin, latitude, and longitude. These fields have had their original values replaced by the k-anonymous equivalents for their grouping. The remaining fields are present for internal use and in the event of a LBS provider returning a result, to link the anonymous query response with the correct user.

Each query contains its own policy values that are parsed from the message by the *Anonymizer* so there is no need to call the Anonymization Service with different arguments per message. Global values such as spatial region boundaries and row/col

```
Latitude1 = 36.92940
Latitude2 = 38.30810
Longitude1 = -123.001200
Longitude2 = -121.217000

#NOTE latitude corresponds to rows a.k.a. y and longitude

#How to split this spatial region into quadrants
#if more than one is specified then the order of
#precedence is
# row_count, col_count
# --row_distance, col_distance
# --May want to support quadrant_area, quadrant_count

#Specify the number of rows and cols the grid supports
Row_Count = 358
Col_Count = 270
```

Figure 4.5: Global Configuration

number are set in a configuration file. Figure 4.5 illustrates the mentioned configurations. Changing these values while the Anonymization Service is running will cause indeterminate errors as the user's coming in after the changes will have a different spatial domain than those already present. Privacy erosion may occur from improperly formed k-anonymous groups as result of the old spatial domain users being combined with users from the new spatial domain.

Chapter 5

EVALUATION

We evaluated the performance of the Anonymization Service on simulated user data. The data is based upon the Mill Avenue demographic data provided by the City of Tempe in [20; 21] as well as movement traces generated by Thomas Brinkhoff's simulator [7; 6]. The box is a 64-bit Windows 7 Professional SP 1 Intel Core i7-2620M @ 2.70 GHz with 4 GB of RAM.

A single large set of movement traces (100,000) was generated with Brinkhoff's simulator in sections of about 10,000 that were combined. Demographic data of Mill Avenue district was then incorporated into the traces, yielding a large set of user movement and demographic information. The policy values were normally distributed with a standard deviation of one about three for k and 36 for mbr. The policy values are modified at runtime by the addition of an test specific offset to the initial value. A total of 49 tests were performed such that every value of $k_\mu \in \{3, 4, 5, 6, 7, 10\}$ was paired with every value of $mbr_\mu \in \{36, 72, 108, 144, 180, 216, 252\}$. The mean and standard deviations are derived from the union of above stated test resultants. From the combined data set we evaluate the affect of various policy pairings on time,Tables A.5 and A.6, as well as information loss, Tables A.1, A.3, A.2, and A.4.

5.1 Affects of Policy on Time

The nature of LBS necessitates a rapid turnaround time from query to result, we have settled on five seconds an allowable delay threshold. Figure 5.1 shows the mean turnaround time for queries with the given k,mbr pair on a linear scale while

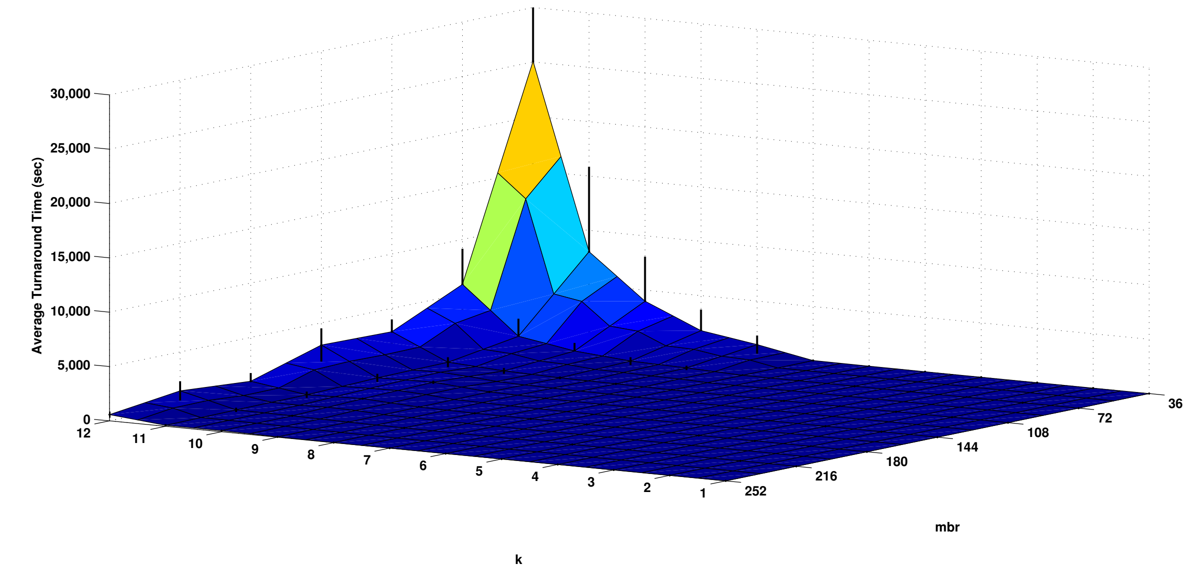

Figure 5.1: Average Turnaround Time (sec)

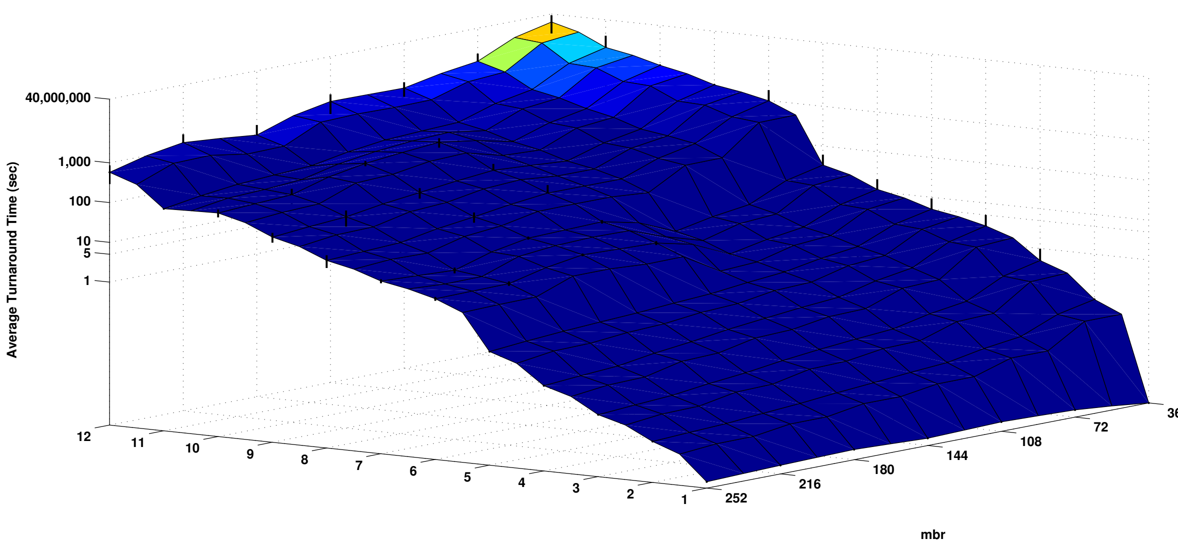

Figure 5.2: Log Average Turnaround Time (sec)

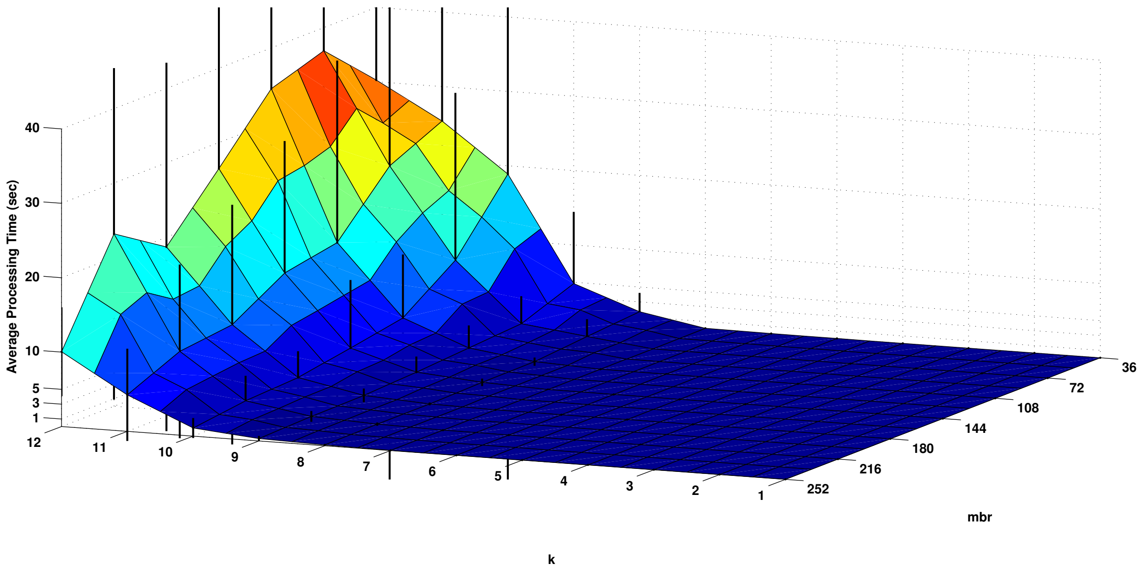

Figure 5.3: Average Processing Time (ms)

Figure 5.2 is on a logarithmic scale. We observe in Figure 5.1 that turnaround time grows exponentially longer as the *mbr* approaches its minimum value of 36. As the k value increases we again see an exponential growth in turnaround time. Figure 5.2 illustrates that k values up to seven with the largest *mbr* and k up to five for the smallest *mbr* meet the turnaround time QoS threshold of five seconds. We evaluate the components, processing and waiting time, of turnaround time next to determine the influence of each on the overall time taken.

5.1.1 Affects of Policy on Processing Time

Processing time, generating a generalization once enough users are present, is dependent upon algorithm used for generalization. Figure 5.3 illustrates that k values up to 11 with the largest *mbr* and k up to seven for the smallest *mbr* meet the turnaround time QoS threshold of five seconds. The k value has greater influence over the processing time than *mbr*; processing time is exponentially related to k. We can also see that on average in the worst case the processing time was under 40

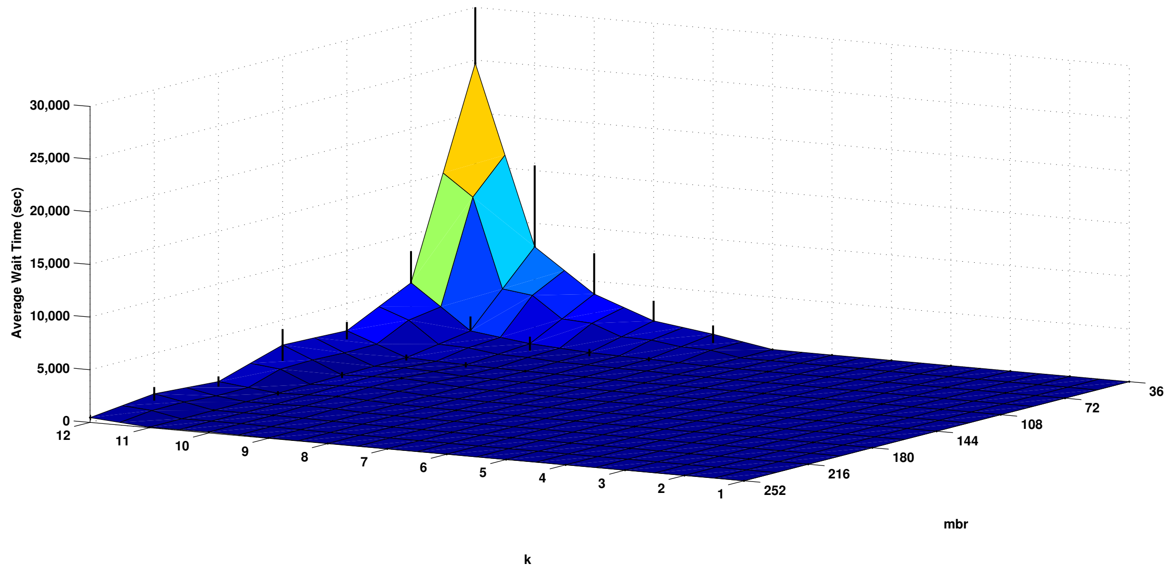

Figure 5.4: Average Wait Time (sec)

seconds while the turnaround time in an equivalent scenario is around 25,000 seconds. Overall the contribution of processing time to turnaround is minuscule.

5.1.2 Affects of Policy on Waiting Time

Waiting time, waiting on enough users to be present for generalization, is dependent upon the Anonymization Service's environment, specially the rate at which users fall into each other *mbr*. Figure 5.4 shows the mean wait time for queries with the given k,*mbr* pair on a linear scale while Figure 5.5 is on a logarithmic scale. We observe in Figure 5.4 that turnaround time grows exponentially longer as the *mbr* approaches its minimum value of 36. As the k value increases we again see an exponential growth in wait time. Figure 5.5 illustrates that k values up to seven with the largest *mbr* and k up to five for the smallest *mbr* meet the turnaround time QoS threshold of five seconds. We can also see that on average in the worst case the waiting time was around 25,000 seconds while the turnaround time in an equivalent scenario is around 25,000 seconds. Waiting time has a major effect on the overall

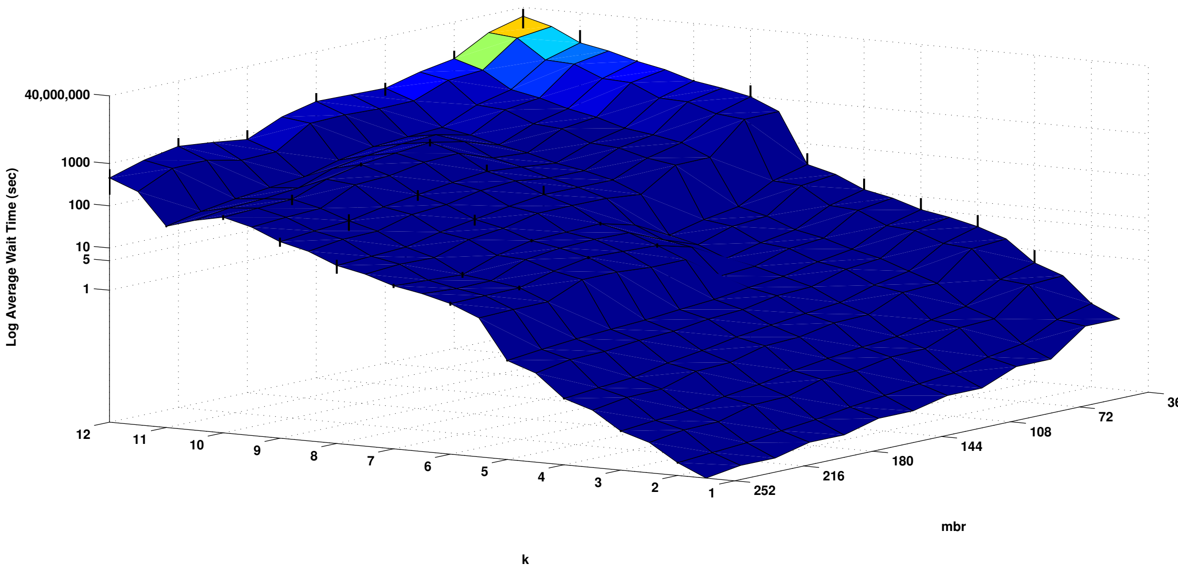

Figure 5.5: Log Average Wait Time (sec)

turnaround time.

5.2 Affects of Policy on Information Loss

In current LBS k-anonymous systems the information released to the LBS provider is quasi-identifier of location. The proof of concept is capable of releasing location as well as other demographic information. The graphs presented below show the amount of information lost by generalization for the given k,mbr pairs. If all quasi-identifier information except location is suppressed then information loss is .66. We observe in Figure 5.6 that information loss grows to an asymptote exponentially as k increases linearly. While information loss grows linearly as mbr grows linearly. As the user is composition of differing quasi-identifiers types we will next examine the effects of policy on NLBS and LBS.

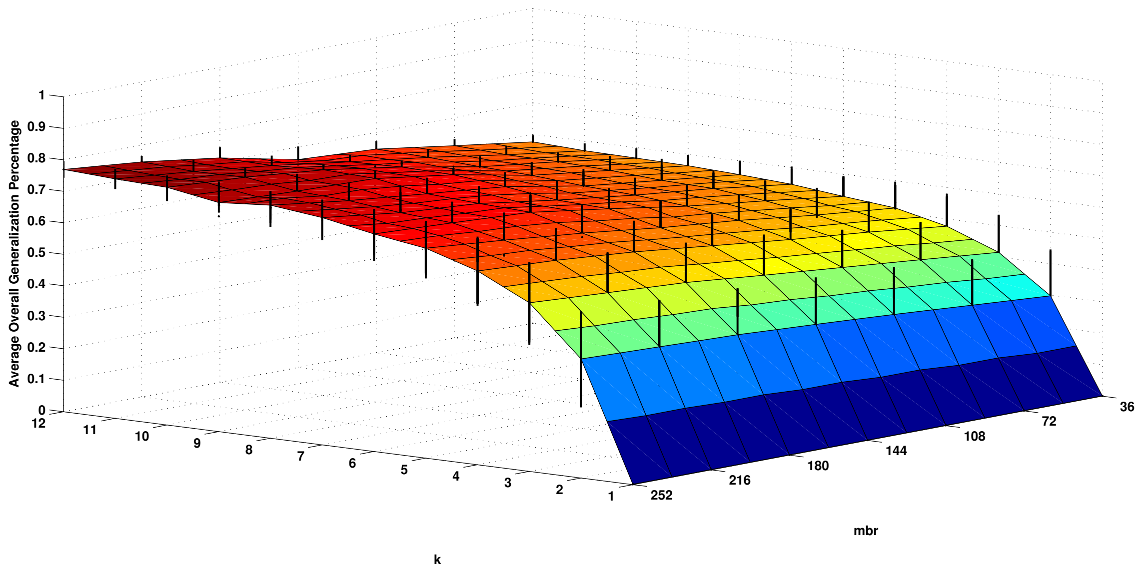

Figure 5.6: Average Overall Percent Generalized

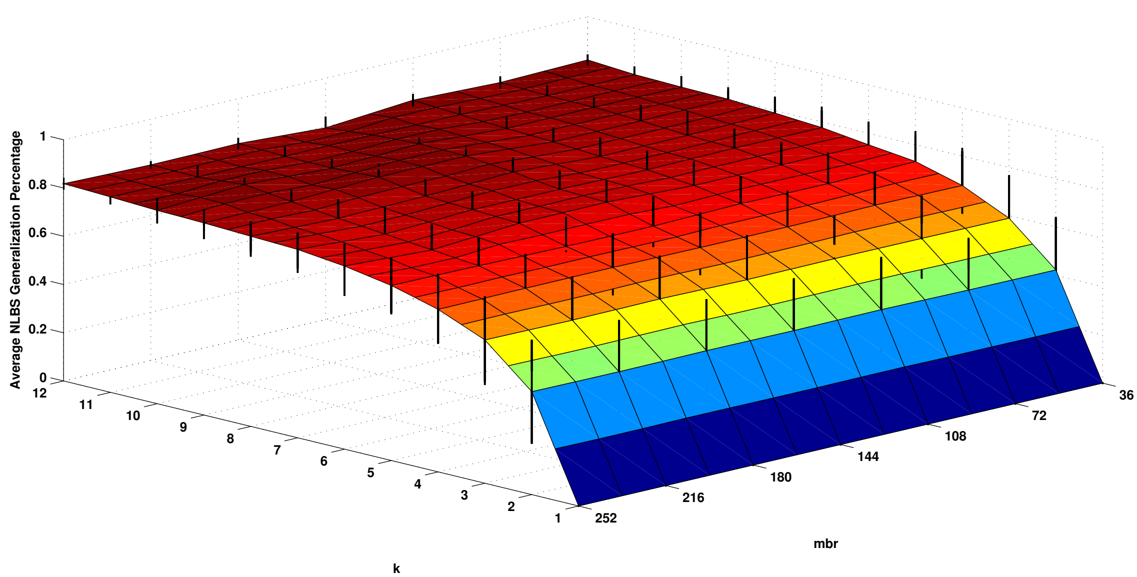

Figure 5.7: Average NLBS Percent Generalized

5.2.1 Effects of Policy on NLBS Information Loss

The NLBS information of the user is composed of *age*, *gender*, *origin*, and *race*, Figures 5.8 through 5.11 respectively. These graphs all grow exponentially to an asymptote as k grows linearly and show no change as the *mbr* varies. The combined effect is presented in Figure 5.7 that also follows this trend. This signifies that demographic information preservation is not dependent upon the *mbr* policy.

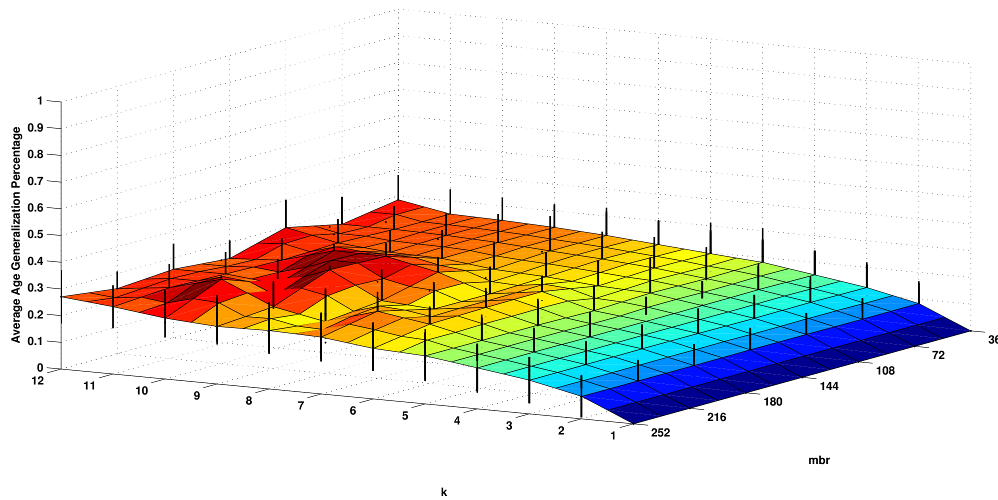

Figure 5.8: Average Age Percent Generalized

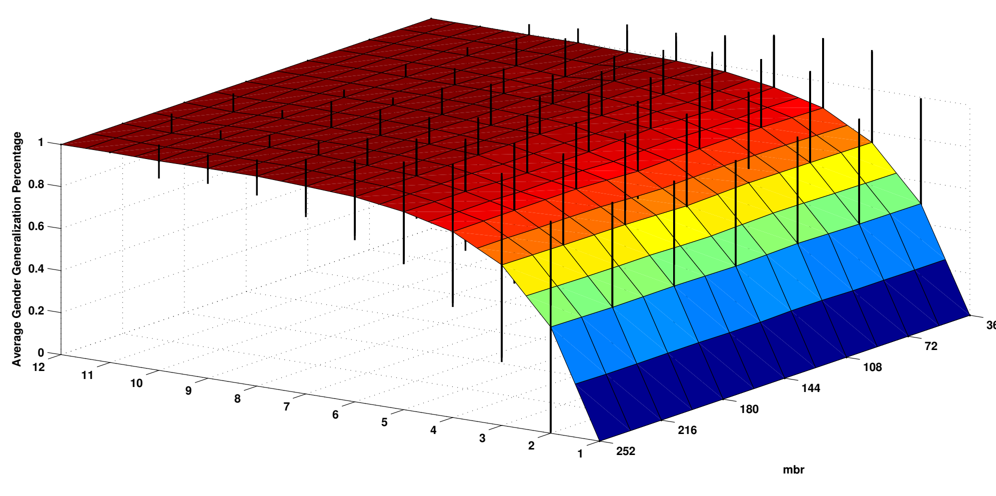

Figure 5.9: Average Gender Percent Generalized

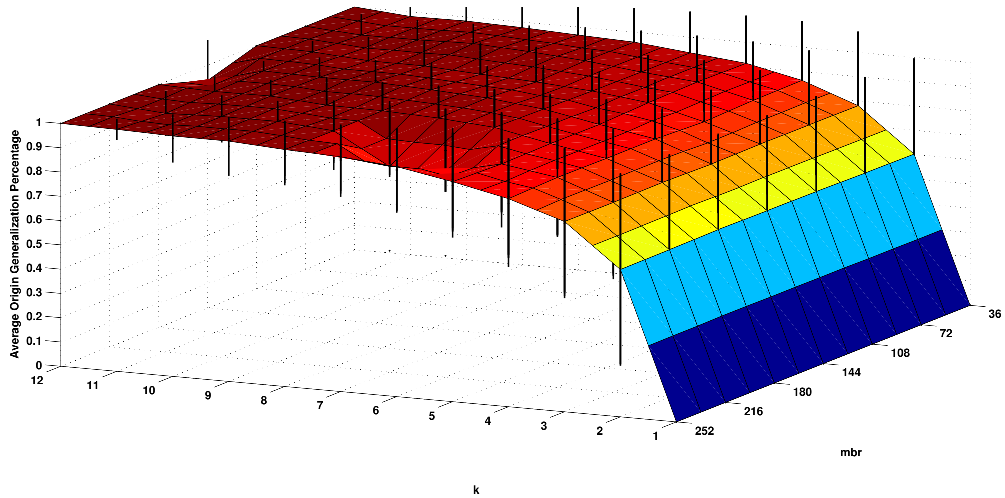

Figure 5.10: Average Origin Percent Generalized

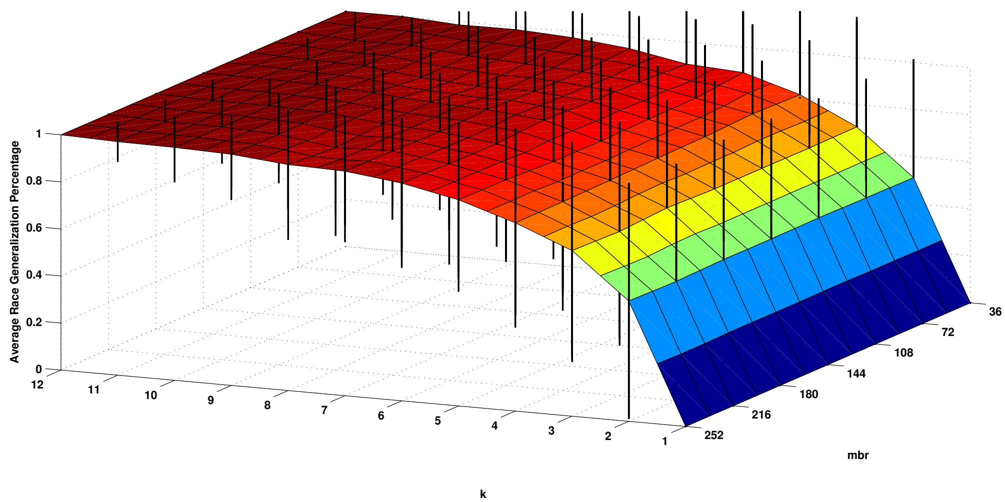

Figure 5.11: Average Race Percent Generalized

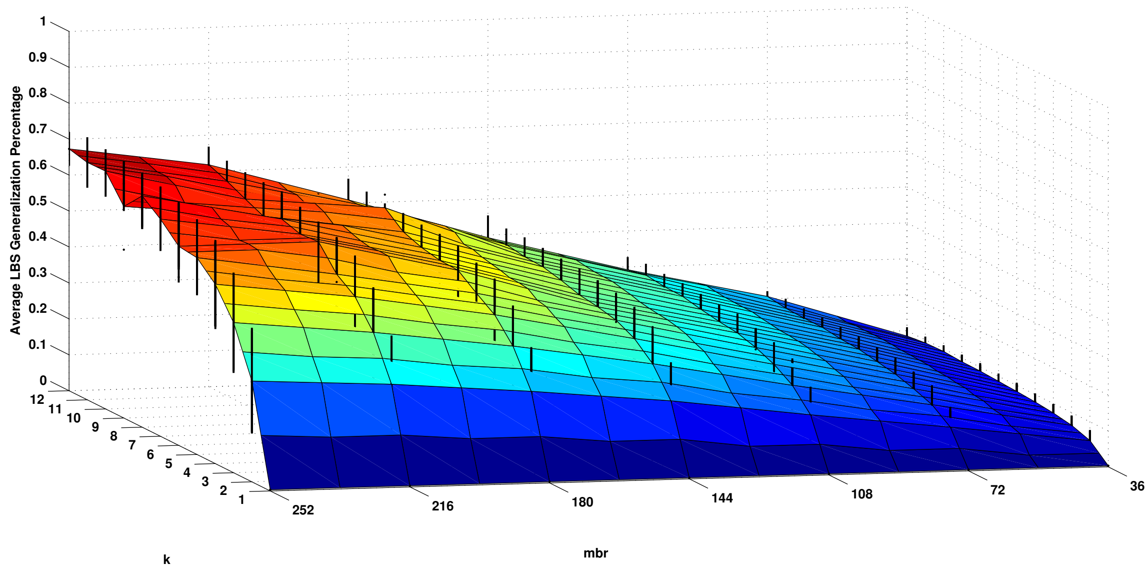

Figure 5.12: Average LBS Percent Generalized

5.2.2 Effects of Policy on LBS Information Loss

The LBS information of the user is composed of *latitude* and *longitude* Figures 5.13 through 5.14 respectively. These graphs all grow exponentially to an asymptote as k grows linearly and shows linear growth as the *mbr* grows. The combined effect is presented in Figure 5.12 that also follows this trend. This signifies that location information preservation is dependent upon both policy values. The *mbr* policy field limits the maximum spatial area of a group hence the asymptotic nature with respect to k, limiting the *mbr* value has a greater effect on the location information loss than changes k.

5.3 Discussion

The approach suffers a performance hit as the number of waiting users increases; primarily due to the underlying graph data structure. As the number of users waiting increases the nodes present in the graph also increasing leading to processing overhead

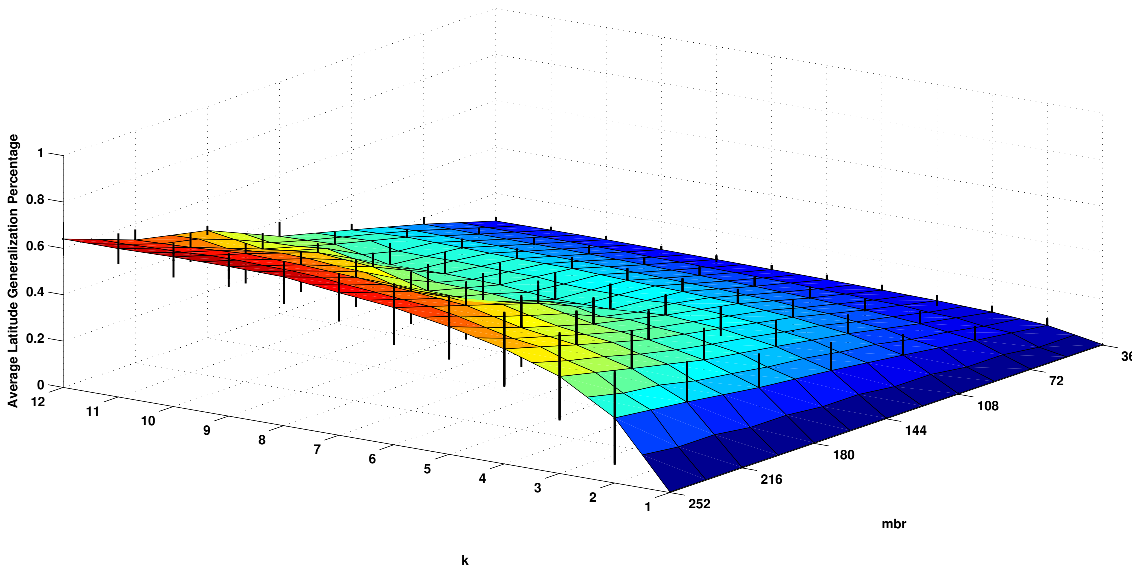

Figure 5.13: Average Latitude Percent Generalized

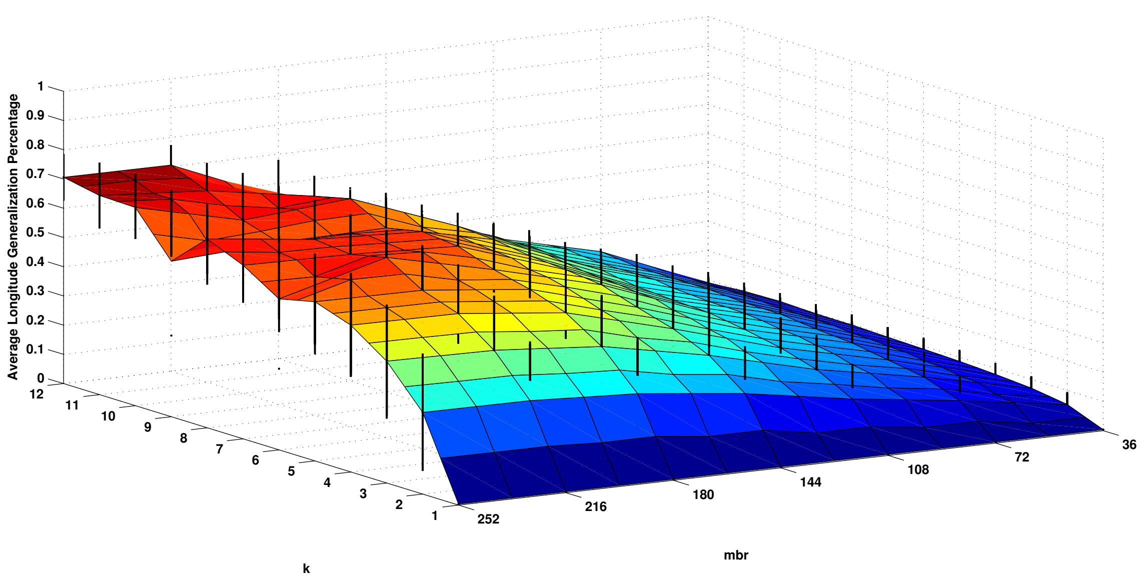

Figure 5.14: Average Longitude Percent Generalized

for the graph dependent operations throughout our algorithm. Implementing an expire time for a message would help to solve this problem as well as communicating the anonymization fail to the user. Along the performance limitations vein is the use of A*, as in the worst case it will end up brute-forcing the solution.

We implicitly trust the user and as such the Anonymization Service is vulnerable to crafted message attacks possibly resulting in erosion of privacy and or denial of service. If an attacker was to flood the Anonymization Service with messages with relatively small k and large mbr they will deny the anonymization of legitimate request because the Anonymization Service. If an attacker was to flood the service with messages with demographic D and message Content C any resultant of the Anonymizer can be used to infer more information about legitimate traffic than expected. This is possible as the attacker can compare their input for a given time period with the anonymized output of the Anonymization Service remove all those with C and infer what would be required to make D match the demographic of the output of the Anonymization Service.

Regardless of the pitfalls of the approach it is able to handle the combined nature of M-Commerce Anonymization meeting time QoS while allowing for up to 80% privacy. We attempted to use open source k-anonymization package Incognito described in [16] but it failed to satisfactorily solve the problem. Failure due to processing time or inability to handle quasi-identifiers with more than a single generalization path.

Chapter 6

CONCLUSION

We have designed and implemented a proof concept Anonymization Service that unifies traditionally disparate approaches for LBS and NLBS information in the LBS domain. The Anonymization Service utilizes a generic approach towards generalization that is not dependent upon the hierarchical structure of NLBS quasi-identifier domains nor the well-ordered structure of LBS quasi-identifier domains. We used a difference measure and single source clique generation to attain k-anonymity. Currently the Anonymization Service supports the NLBS quasi-identifiers of *race*, *age*, *gender*,and *origin*; the LBS quasi-identifiers of *longitude* and *latitude*; the policy values of k and *mbr*. The increased expressive power of generic view of PII allows for domains such *race* to be represented more accurately, as an individual may have more than a single race in their immediate heritage. It also allows for the quasi-identifiers regardless of expression; hierarchy, line, or set, to be processed simultaneously.

The generic view of personally identifiable information (PII) as quasi-identifiers as well as the Anonymization Service required to process this view are our contributions. Namely we have designed the following:

1. User-Centric Anonymization algorithm separated from the quasi-identifier domains it handles

2. Difference measure as comparator of people

3. Set based quasi-identifier domains

4. Methodologies for working with Set based quasi-identifier domains

5. Quasi-identifier Domain abstraction/interface

6. User policy unification technique

The implementation of some of the aforementioned designs requires submodules so the implementation list below does not directly reflect the design list above.

1. User-Centric Anonymization algorithm separated from the quasi-identifier domains it handles (*Anonymizer*)

2. Difference measure as comparator of people (*Anonymizer*)

3. Quasi-identifier Domain abstraction/interface (*Domains*)

4. User policy unification technique for k and mbr (*Enforcers*)

5. Jgrapht based Hierarchical or partially order quasi-identifier domain module. (*Domains*)

6. List based well-ordered quasi-identifier domain module. (*Domains*)

6.1 Future work

Regarding future work, the unordered set can be implemented into the Anonymization Service. A Framework for the handling for domains may be developed to support the Anonymization Service in generalizing the varied domains and the LBS provider process the anonymous queries composed of the varied domains. The current implemented Anonymization Service can be extended to include LBS provider policy constraints and l-diversity [18] and t-closeness [17] of query content. As the mbr

policy has little influence on the amount of NLBS information lost a different *time-window* approach may be taken that is presented in [19]. The approach increases the diversity of users available by retaining a sliding history of users in a given region, the increased diversity should give Anonymization Service more choices within the same region and yield k-anonymous groups with less NLBS information loss.

Beyond performance improvements a distribution and lookup service for the various Generalization Domains needs to implemented. Information required tends to be service type specific so the number of Generalization domains is expected to grow quite large. Aside from the Anonymization Service research needs to continue on the how to best present the policy management to users, educating them on the expected degradation of QoS given their selected policy values.

REFERENCES

[1] “Jgrapht”, URL `http://jgrapht.org/` (2013).

[2] Aggarwal, C. C., “On k-anonymity and the curse of dimensionality”, in “Proceedings of the 31st international conference on Very large data bases”, pp. 901–909 (VLDB Endowment, 2005).

[3] Aggarwal, G., T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas and A. Zhu, “Anonymizing tables”, in “Database Theory-ICDT 2005”, pp. 246–258 (Springer, 2005).

[4] Bamba, B., L. Liu, P. Pesti and T. Wang, “Supporting anonymous location queries in mobile environments with privacygrid”, in “Proceedings of the 17th International Conference on World Wide Web”, WWW ’08, pp. 237–246 (ACM, New York, NY, USA, 2008), URL `http://doi.acm.org/10.1145/1367497.1367531`.

[5] Bayardo, R. J. and R. Agrawal, “Data privacy through optimal k-anonymization”, in “Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on”, pp. 217–228 (IEEE, 2005).

[6] Brinkhoff, T., “A framework for generating network-based moving objects”, GeoInformatica **6**, 2, 153–180 (2002).

[7] Brinkhoff, T., “Thomas brinkhoff: Network-based generator of moving objects”, URL `http://iapg.jade-hs.de/personen/brinkhoff/generator/` (2012).

[8] Ciriani, V., S. De Capitani di Vimercati, S. Foresti and P. Samarati, “k-anonymity”, in “Secure Data Management in Decentralized Systems”, edited by T. Yu and S. Jajodia (Springer-Verlag, 2007), URL `http://spdp.di.unimi.it/papers/k-Anonymity.pdf`.

[9] Gedik, B. and L. Liu, “A customizable k-anonymity model for protecting location privacy”, (2004).

[10] Gedik, B. and L. Liu, “Location privacy in mobile systems: A personalized anonymization model”, in “Proceedings of the 25th IEEE International Conference on Distributed Computing Systems”, ICDCS ’05, pp. 620–629 (IEEE Computer Society, Washington, DC, USA, 2005), URL `http://dx.doi.org/10.1109/ICDCS.2005.48`.

[11] Gedik, B. and L. Liu, “Protecting location privacy with personalized k-anonymity: Architecture and algorithms”, IEEE Transactions on Mobile Computing **7**, 1, 1–18, URL `http://dx.doi.org/10.1109/TMC.2007.1062` (2008).

[12] Ghinita, G., P. Kalnis and S. Skiadopoulos, “Mobihide: a mobilea peer-to-peer system for anonymous location-based queries”, in “Advances in Spatial and Temporal Databases”, pp. 221–238 (Springer, 2007).

[13] Gruteser, M. and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking", in "Proceedings of the 1st International Conference on Mobile Systems, Applications and Services", MobiSys '03, pp. 31–42 (ACM, New York, NY, USA, 2003), URL http://doi.acm.org/10.1145/1066116.1189037.

[14] Kalnis, P., G. Ghinita, K. Mouratidis and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries", IEEE Trans. on Knowl. and Data Eng. **19**, 12, 1719–1733, URL http://dx.doi.org/10.1109/TKDE.2007.190662 (2007).

[15] Lee, R., "Cell phone ownership hits 91% of adults", URL http://www.pewresearch.org/fact-tank/2013/06/06/cell-phone-ownership-hits-91-of-adults/ (2013).

[16] LeFevre, K., D. J. DeWitt and R. Ramakrishnan, "Incognito: Efficient full-domain k-anonymity", in "Proceedings of the 2005 ACM SIGMOD international conference on Management of data", pp. 49–60 (ACM, 2005).

[17] Li, N., T. Li and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity.", in "ICDE", vol. 7, pp. 106–115 (2007).

[18] Machanavajjhala, A., D. Kifer, J. Gehrke and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity", ACM Transactions on Knowledge Discovery from Data (TKDD) **1**, 1, 3 (2007).

[19] Masoumzadeh, A., J. Joshi and H. A. Karimi, "Lbs (k, t)-anonymity: A spatio-temporal approach to anonymity for location-based service users", in "Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems", GIS '09, pp. 464–467 (ACM, New York, NY, USA, 2009), URL http://doi.acm.org/10.1145/1653771.1653845.

[20] of Tempe Community Development, T. C., "Demographics part 1", URL http://www.millavenue.com/_files/docs/demographicdetailjune2009pg-17.pdf (2009).

[21] of Tempe Community Development, T. C., "Demographics part 2", URL http://www.millavenue.com/_files/docs/demographic-detail-june-2009-pg-8-13.pdf (2009).

[22] Samarati, P., "Protecting respondents identities in microdata release", Knowledge and Data Engineering, IEEE Transactions on **13**, 6, 1010–1027 (2001).

[23] Samarati, P. and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression", Tech. rep., Technical report, SRI International (1998).

[24] Sweeney, L., "Achieving k-anonymity privacy protection using generalization and suppression", International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **10**, 05, 571–588 (2002).

[25] Sweeney, L., “k-anonymity: A model for protecting privacy”, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **10**, 05, 557–570 (2002).

APPENDIX A

DATA COLLECTED FROM EXPERIMENTATION ON SIMULATED USER BASE

mobihide [12]

Table A.1: Average NLBS Generalization Percentage

k	mbr	Gender	Origin	Race	Age	NLBS
1	36	0	0	0	0	0
1	72	0	0	0	0	0
1	108	0	0	0	0	0
1	144	0	0	0	0	0
1	180	0	0	0	0	0
1	216	0	0	0	0	0
1	252	0	0	0	0	0
2	36	0.493888996	0.602123812	0.510543971	0.083670074	0.422556713
2	72	0.496751733	0.603997764	0.511900635	0.083332129	0.423995565
2	108	0.504160878	0.607513322	0.514776281	0.084087181	0.427634415
2	144	0.495794393	0.609064532	0.512699757	0.084445505	0.425501047
2	180	0.501270498	0.611254	0.51289956	0.083613209	0.427259317
2	216	0.49970548	0.607994152	0.507301377	0.08363392	0.424658732
2	252	0.505938109	0.606273366	0.511334336	0.084112286	0.426914524
3	36	0.747693902	0.780715011	0.704879337	0.12487615	0.5895411
3	72	0.75149294	0.781563756	0.702294848	0.127406732	0.590689569
3	108	0.754621299	0.782584416	0.703353885	0.125354512	0.591478528
3	144	0.745835035	0.780147589	0.703348344	0.127073885	0.589101213
3	180	0.755825821	0.78419798	0.710146375	0.127419016	0.594397298
3	216	0.758615367	0.782143351	0.697633992	0.12633935	0.591183015
3	252	0.763678992	0.783166614	0.703915973	0.125930404	0.594172996
4	36	0.874408651	0.863855001	0.819491176	0.153787061	0.677885472
4	72	0.881070115	0.86130582	0.825193105	0.156256351	0.680956348
4	108	0.879739283	0.853532104	0.815824338	0.154541256	0.675909245
4	144	0.872680124	0.862702713	0.812250212	0.153683398	0.675329112
4	180	0.887461665	0.862284643	0.806430018	0.153247547	0.677355968
4	216	0.886194714	0.864255622	0.802000968	0.154449627	0.676725233
4	252	0.884609457	0.856342684	0.800011233	0.15352642	0.673622448
5	36	0.935055305	0.913516062	0.891950581	0.181861923	0.730595968
5	72	0.926387728	0.916399222	0.870275835	0.178914559	0.722994336
5	108	0.938130363	0.91135122	0.868398775	0.18263147	0.725127957
5	144	0.927611259	0.901643433	0.862902254	0.17483903	0.716748994
5	180	0.926928375	0.902709232	0.863695907	0.178806291	0.718034951
5	216	0.937969431	0.898626251	0.873641687	0.179278807	0.722379044
5	252	0.941698484	0.905740109	0.875890194	0.179876228	0.725801254
6	36	0.971963544	0.935874205	0.906523942	0.195815245	0.752544234
6	72	0.963284441	0.936720405	0.916721209	0.200067008	0.754198266
6	108	0.962212062	0.938200719	0.922278728	0.191245876	0.753484346
6	144	0.973938851	0.943944397	0.930931031	0.228160491	0.769243693
6	180	0.974223326	0.948361963	0.918386032	0.226329878	0.7668253

Continued on next page

Table A.1 – *Continued from previous page*

k	mbr	Gender	Origin	Race	Age	NLBS
6	216	0.974677076	0.846983136	0.920152913	0.221673032	0.740871539
6	252	0.974919942	0.947068641	0.925416601	0.189126137	0.75913283
7	36	0.98624807	0.959078935	0.950426094	0.211227202	0.776745075
7	72	0.990130077	0.961961816	0.939647714	0.214700235	0.776609961
7	108	0.981181796	0.954894789	0.943771269	0.206414587	0.77156561
7	144	0.978286071	0.969164716	0.953713712	0.200253367	0.775354467
7	180	0.988707352	0.966339046	0.946567765	0.201762445	0.775844152
7	216	0.987200805	0.847266194	0.952533922	0.251402713	0.759600909
7	252	0.988179096	0.966912716	0.954485896	0.212697176	0.780568721
8	36	0.989310237	0.967420006	0.97500961	0.226941804	0.789670414
8	72	0.972440244	0.97900065	0.951234623	0.22669132	0.782341709
8	108	0.993114189	0.965959539	0.94305225	0.215730854	0.779464208
8	144	0.992521685	0.981723015	0.963122174	0.263360858	0.800181933
8	180	0.99242818	0.975227981	0.959362597	0.214308173	0.785331733
8	216	0.998763906	0.973062746	0.945481684	0.193980023	0.77782209
8	252	0.998148148	0.97549264	0.960600665	0.21850623	0.788186921
9	36	0.997470489	0.977146571	0.98777403	0.238346648	0.800184435
9	72	0.986918336	0.991525424	0.963836672	0.239382428	0.795415715
9	108	0.996551724	0.979177719	0.958222812	0.238256935	0.793052297
9	144	0.999533147	0.992063492	0.981064426	0.303834579	0.819123911
9	180	0.992927284	0.977803037	0.985412522	0.311755963	0.816974702
9	216	0.998756219	0.99063644	0.979336807	0.231170579	0.799975011
9	252	0.998347107	0.98953168	0.982691408	0.215140937	0.796427783
10	36	0.997356828	0.983700441	0.985903084	0.246182454	0.803285702
10	72	0.998392283	0.986334405	0.985530547	0.247604582	0.804465454
10	108	0.996363636	0.98	0.987272727	0.242256738	0.801473275
10	144	0.999271137	0.991618076	0.994169096	0.306602253	0.822915141
10	180	0.999174917	0.994636964	0.992574257	0.240078279	0.806616104
10	216	0.999019608	0.990196078	0.989215686	0.315435018	0.823466598
10	252	0.996726678	0.989770867	0.990180033	0.227680958	0.801089634
11	36	1	0.981632653	1	0.250969131	0.808150446
11	72	1	0.984962406	0.984962406	0.251445196	0.805342502
11	108	1	0.995867769	0.991735537	0.258565555	0.811542215
11	144	1	0.996774194	1	0.252034365	0.81220214
11	180	0.992125984	0.992125984	0.992125984	0.258576346	0.808738575
11	216	0.991150442	0.982300885	0.991150442	0.265486726	0.807522124
11	252	1	0.996268657	0.992537313	0.251856296	0.810165567
12	36	1	1	1	0.2860134	0.82150335
12	72	1	1	1	0.252261307	0.813065327
12	108	1	1	1	0.298157454	0.824539363
12	144	1	0.9375	1	0.239949749	0.794362437
12	180	1	1	1	0.257717157	0.814429289

Continued on next page

Table A.1 – *Continued from previous page*

k	mbr	Gender	Origin	Race	Age	NLBS
12	216	1	1	1	0.24321608	0.81080402
12	252	1	1	1	0.270638909	0.817659727

Table A.2: Average STD of NLBS Generalization Percentage

k	mbr	Gender	Origin	Race	Age	NLBS
1	36	0	0	0	0	6.29E-16
1	72	0	0	0	0	6.30E-16
1	108	0	0	0	0	6.30E-16
1	144	0	0	0	0	6.29E-16
1	180	0	0	0	0	6.30E-16
1	216	0	0	0	0	6.31E-16
1	252	0	0	0	0	6.31E-16
2	36	0.499959537	0.392863972	0.499867678	0.078404718	0.144227903
2	72	0.499987785	0.393471838	0.499804963	0.077463929	0.143790331
2	108	0.499933982	0.392226635	0.499722908	0.077588807	0.144970911
2	144	0.499938882	0.391223936	0.499755859	0.07894809	0.146128812
2	180	0.499997522	0.39179699	0.499780379	0.078393062	0.146798634
2	216	0.49999698	0.391913579	0.499945247	0.07774391	0.148687942
2	252	0.499945812	0.391803053	0.499867387	0.078739465	0.150740119
3	36	0.434231024	0.29957524	0.45585525	0.087840234	0.118121697
3	72	0.432081899	0.300410314	0.457245346	0.08727228	0.118869844
3	108	0.430231986	0.302068188	0.456776561	0.08505019	0.119136515
3	144	0.435363313	0.305510494	0.456751332	0.086672851	0.123455048
3	180	0.429503585	0.300625776	0.453593266	0.086173058	0.123119112
3	216	0.427656383	0.301408911	0.459113374	0.08659315	0.125758227
3	252	0.423769889	0.302394918	0.456405595	0.086378171	0.126523495
4	36	0.331357916	0.238352696	0.384041066	0.089542341	0.096239235
4	72	0.322022646	0.23761377	0.378607507	0.089805839	0.092856598
4	108	0.324930887	0.246488493	0.38734144	0.088332348	0.097325386
4	144	0.333316153	0.241547278	0.390293564	0.08929902	0.098256831
4	180	0.314159655	0.239613699	0.395071837	0.088555564	0.097508958
4	216	0.315451007	0.239386821	0.397742782	0.08774691	0.101356739
4	252	0.318727036	0.246027065	0.399627549	0.088440369	0.104799312
5	36	0.246318863	0.192828476	0.307419983	0.092852646	0.075126193
5	72	0.258396246	0.18890261	0.334404697	0.094039465	0.07982684
5	108	0.240835548	0.19511258	0.337669374	0.092555036	0.077855439
5	144	0.255972051	0.207868413	0.34150829	0.086662698	0.084131746
5	180	0.257396993	0.20155078	0.342119897	0.090815846	0.084633976
5	216	0.240690785	0.210061444	0.332137427	0.088674044	0.084340724
5	252	0.234020692	0.205856378	0.328890313	0.089094666	0.084221529

Continued on next page

Table A.2 – *Continued from previous page*

k	mbr	Gender	Origin	Race	Age	NLBS
6	36	0.164264163	0.168831346	0.289948425	0.093023341	0.062324932
6	72	0.185634923	0.168667871	0.275774739	0.092165754	0.063758592
6	108	0.188637268	0.166016848	0.266643131	0.089718732	0.064474732
6	144	0.141770014	0.144772862	0.224432173	0.07156474	0.05666156
6	180	0.140936711	0.135019263	0.238852447	0.071161645	0.057031818
6	216	0.139665781	0.136589578	0.236873268	0.071172293	0.055601981
6	252	0.138976738	0.136909479	0.231849567	0.071418766	0.057767597
7	36	0.114231724	0.136426364	0.216390885	0.090107704	0.051031576
7	72	0.08043713	0.132218463	0.237022653	0.090568085	0.05193208
7	108	0.135808735	0.143704004	0.230306275	0.087840671	0.054189224
7	144	0.11759354	0.102478827	0.179524591	0.068732772	0.045358296
7	180	0.090965555	0.107133228	0.191283543	0.066173256	0.043139484
7	216	0.096998671	0.107234463	0.182109834	0.071681622	0.048572944
7	252	0.092964731	0.106241442	0.178034214	0.066524035	0.044312567
8	36	0.082742913	0.123250215	0.155745408	0.093596827	0.041803316
8	72	0.159312328	0.099451175	0.214569609	0.090047916	0.054489066
8	108	0.06328785	0.125410695	0.224525156	0.08758222	0.047844186
8	144	0.059767653	0.080112018	0.160859621	0.065904665	0.036912009
8	180	0.060475549	0.092707675	0.167442871	0.064918896	0.037581206
8	216	0.017535551	0.096285522	0.188225583	0.064883776	0.039982574
8	252	0.021436735	0.092262404	0.163622608	0.06437068	0.036543825
9	36	0.035473329	0.104312052	0.077223769	0.085588906	0.030675744
9	72	0.109132281	0.045241857	0.173912955	0.096594439	0.040152052
9	108	0.04137931	0.099808244	0.195942208	0.095676792	0.04131766
9	144	0.012465948	0.035493143	0.10816863	0.065732807	0.023351211
9	180	0.059581561	0.082723601	0.097392453	0.060989429	0.027389251
9	216	0.020323551	0.054264545	0.114797536	0.059619423	0.028759724
9	252	0.023414357	0.040436793	0.104948735	0.057830998	0.028720768
10	36	0.051343797	0.088792478	0.117890599	0.086122006	0.031087411
10	72	0.040064102	0.081523303	0.119415612	0.082676277	0.029216422
10	108	0.060192529	0.09797959	0.112095001	0.08210199	0.032823601
10	144	0.01907617	0.045002493	0.053679162	0.043406695	0.016642994
10	180	0.020294346	0.036221498	0.060479167	0.04444942	0.015821227
10	216	0.022118655	0.048526936	0.072635084	0.042806572	0.02018784
10	252	0.040323028	0.049524216	0.069379766	0.044545739	0.0203557
11	36	0	0.094054846	0	0.093070062	0.023192918
11	72	0	0.085397118	0.121702361	0.088066517	0.028230332
11	108	0	0.045266327	0.090532654	0.089994272	0.026132452
11	144	0	0.040031205	0	0.082608999	0.020866786
11	180	0.088385608	0.06224956	0.088385608	0.082385024	0.031485405
11	216	0.093654914	0.092392093	0.093654914	0.088205552	0.036853117
11	252	0	0.04303195	0.0860639	0.080680546	0.032165321

Continued on next page

Table A.2 – *Continued from previous page*

k	mbr	Gender	Origin	Race	Age	NLBS
12	36	0	0	0	0.092669147	0.021479627
12	72	0	0	0	0.102856846	0.020953404
12	108	0	0	0	0.103866104	0.025485034
12	144	0	0.165359457	0	0.068152543	0.04148346
12	180	0	0	0	0.096012942	0.033783807
12	216	0	0	0	0.064556731	0.019583789
12	252	0	0	0	0.100014197	0.026502434

Table A.3: Average LBS Generalization Percentage

k	mbr	Latitude	Longitude	LBS	Overall
1	36	0.002793296	0.003703704	0.0032485	0.001082833
1	72	0.002793296	0.003703704	0.0032485	0.001082833
1	108	0.002793296	0.003703704	0.0032485	0.001082833
1	144	0.002793296	0.003703704	0.0032485	0.001082833
1	180	0.002793296	0.003703704	0.0032485	0.001082833
1	216	0.002793296	0.003703704	0.0032485	0.001082833
1	252	0.002793296	0.003703704	0.0032485	0.001082833
2	36	0.044179411	0.051990198	0.048084805	0.297732744
2	72	0.085197454	0.107779185	0.096488319	0.314826483
2	108	0.128144811	0.155699124	0.141921968	0.332396933
2	144	0.167093188	0.216943497	0.192018342	0.347673478
2	180	0.199975641	0.259064179	0.22951991	0.361346181
2	216	0.243500424	0.277357023	0.260428724	0.369915396
2	252	0.281044047	0.279808209	0.280426128	0.378085059
3	36	0.060047472	0.070309353	0.065178413	0.414753538
3	72	0.116374541	0.150497975	0.133436258	0.438271798
3	108	0.178994412	0.218487653	0.198741032	0.460566029
3	144	0.229686846	0.309795796	0.269741321	0.482647916
3	180	0.277741323	0.382503617	0.33012247	0.506305689
3	216	0.347074843	0.411635493	0.379355168	0.520573733
3	252	0.418302121	0.416983088	0.417642604	0.535329532
4	36	0.068176977	0.079346798	0.073761888	0.476510944
4	72	0.133872889	0.171199608	0.152536248	0.504816315
4	108	0.208305226	0.255083913	0.231694569	0.527837687
4	144	0.266379127	0.362541082	0.314460105	0.555039443
4	180	0.323770585	0.459986624	0.391878605	0.582196847
4	216	0.407398871	0.499396759	0.453397815	0.60228276
4	252	0.495901162	0.502186517	0.49904384	0.615429579
5	36	0.072683008	0.086224564	0.079453786	0.513548574
5	72	0.144316073	0.191185788	0.167750931	0.537913201
5	108	0.225975636	0.27991296	0.252944298	0.567733404

Continued on next page

Table A.3 – *Continued from previous page*

k	mbr	Latitude	Longitude	LBS	Overall
5	144	0.282534554	0.398098577	0.340316566	0.591271518
5	180	0.350403751	0.50247821	0.42644098	0.620836961
5	216	0.439773074	0.541407073	0.490590074	0.645116054
5	252	0.550022095	0.544650834	0.547336464	0.666312991
6	36	0.076574167	0.09000662	0.083290393	0.52945962
6	72	0.153441808	0.201062606	0.177252207	0.561882913
6	108	0.236745346	0.299676383	0.268210865	0.591726519
6	144	0.266389494	0.437538419	0.351963956	0.630150447
6	180	0.397987613	0.531673535	0.464830574	0.666160391
6	216	0.495090093	0.616008134	0.555549113	0.679097397
6	252	0.591442805	0.516434267	0.553938536	0.690734732
7	36	0.078805957	0.093481253	0.086143605	0.546544585
7	72	0.158782179	0.214482871	0.186632525	0.579950816
7	108	0.243727231	0.305184641	0.274455936	0.605862386
7	144	0.267708025	0.451801852	0.359754938	0.636821291
7	180	0.360566928	0.574081392	0.46732416	0.673004155
7	216	0.51443255	0.588970479	0.551701514	0.690301111
7	252	0.614737456	0.593654049	0.604195752	0.721777732
8	36	0.080890307	0.097297714	0.089094011	0.556144947
8	72	0.164212153	0.215501539	0.189856846	0.584846755
8	108	0.253596979	0.313274541	0.28343576	0.614121392
8	144	0.289192893	0.461465873	0.375329383	0.658564416
8	180	0.368984108	0.586513734	0.477748921	0.682804129
8	216	0.533516671	0.602100976	0.567808823	0.707817668
8	252	0.641328878	0.646261827	0.643795352	0.740056398
9	36	0.082156359	0.097858829	0.090007594	0.563458821
9	72	0.169279596	0.223956229	0.196617913	0.595816448
9	108	0.255178526	0.332841144	0.294009835	0.62670481
9	144	0.295852568	0.478713836	0.387283202	0.675177008
9	180	0.431370344	0.561018785	0.496194564	0.710047989
9	216	0.546412294	0.553997309	0.550204802	0.716718275
9	252	0.64521568	0.532429575	0.588822628	0.727226065
10	36	0.084180346	0.102264643	0.093222495	0.566597966
10	72	0.171203902	0.223490532	0.197347217	0.602092709
10	108	0.26111224	0.330572391	0.295842315	0.632929622
10	144	0.292829861	0.487112623	0.389971242	0.678600508
10	180	0.4521383	0.623288718	0.537713509	0.716981906
10	216	0.55334374	0.592509078	0.572926409	0.739953201
10	252	0.645335973	0.678008123	0.661672048	0.754617105
11	36	0.083741877	0.099168556	0.091455216	0.569252036
11	72	0.17244928	0.225814536	0.199131908	0.603272304
11	108	0.265363128	0.339087848	0.302225488	0.641769973

Continued on next page

Table A.3 – *Continued from previous page*

k	mbr	Latitude	Longitude	LBS	Overall
11	144	0.341070463	0.455937873	0.398504168	0.674302816
11	180	0.426516518	0.594225722	0.51037112	0.709282756
11	216	0.539476937	0.655293346	0.597385142	0.737476463
11	252	0.639769032	0.681260365	0.660514698	0.760281944
12	36	0.081936685	0.088580247	0.085258466	0.576088389
12	72	0.174301676	0.239259259	0.206780468	0.610970374
12	108	0.255586592	0.328395062	0.291990827	0.647023185
12	144	0.333798883	0.450925926	0.392362404	0.660362426
12	180	0.464884278	0.552910053	0.508897165	0.712585248
12	216	0.530167598	0.705185185	0.617676391	0.746428144
12	252	0.640462889	0.705820106	0.673141497	0.769486984

Table A.4: Average STD of LBS Generalization Percentage

k	mbr	Latitude	Longitude	LBS
1	36	2.36E-15	3.18E-15	0
1	72	2.37E-15	3.17E-15	0
1	108	2.36E-15	3.18E-15	0
1	144	2.36E-15	3.17E-15	0
1	180	2.37E-15	3.17E-15	0
1	216	2.36E-15	3.18E-15	0
1	252	2.37E-15	3.18E-15	0
2	36	0.029872769	0.040380919	0.216126451
2	72	0.058154164	0.080425915	0.214374919
2	108	0.087647776	0.114967493	0.213776471
2	144	0.115435818	0.156385702	0.213435266
2	180	0.143352764	0.184449644	0.212402757
2	216	0.173555566	0.196750373	0.213172638
2	252	0.199611685	0.199586712	0.214204675
3	36	0.026946938	0.039592989	0.176971476
3	72	0.053966169	0.077682756	0.176954218
3	108	0.080549079	0.110526731	0.175727251
3	144	0.105157284	0.148006958	0.179603015
3	180	0.135197137	0.170958679	0.17613253
3	216	0.163713361	0.18523251	0.178986108
3	252	0.185536445	0.189428563	0.176600442
4	36	0.023953953	0.037744439	0.144152269
4	72	0.048366956	0.071935502	0.137078878
4	108	0.070187114	0.100531525	0.142612197
4	144	0.092715396	0.130307802	0.141738341
4	180	0.122491008	0.143995236	0.139735294

Continued on next page

Table A.4 – *Continued from previous page*

k	mbr	Latitude	Longitude	LBS
4	216	0.146561895	0.159903158	0.140012176
4	252	0.15784674	0.169058763	0.140792071
5	36	0.022082317	0.036702061	0.112178297
5	72	0.045595212	0.065600938	0.118272143
5	108	0.061255343	0.090356038	0.113459886
5	144	0.086068468	0.11304168	0.119903177
5	180	0.110282355	0.122973127	0.119163981
5	216	0.134295819	0.14337753	0.11634864
5	252	0.131150079	0.150881978	0.113700952
6	36	0.020232234	0.036108148	0.093197121
6	72	0.039038468	0.061394942	0.093897253
6	108	0.056660048	0.082478233	0.09450942
6	144	0.061048345	0.080000233	0.08036765
6	180	0.083926459	0.082118275	0.079201595
6	216	0.093495259	0.105401565	0.078268292
6	252	0.095400047	0.111453511	0.079213956
7	36	0.018954167	0.03371069	0.07640427
7	72	0.037823208	0.052475019	0.076462317
7	108	0.05209244	0.077304396	0.078869383
7	144	0.050467437	0.070701091	0.061779236
7	180	0.072973475	0.066376278	0.060558915
7	216	0.080158177	0.087584662	0.063644927
7	252	0.07313282	0.098590345	0.058140728
8	36	0.016963659	0.033235401	0.061455718
8	72	0.03265702	0.050935363	0.080065356
8	108	0.044223635	0.071633105	0.070794724
8	144	0.047750323	0.064442111	0.051051205
8	180	0.063098999	0.058942651	0.050910856
8	216	0.069705147	0.083132787	0.056543583
8	252	0.061580501	0.089622825	0.049735502
9	36	0.016010949	0.032332932	0.043856421
9	72	0.031618382	0.040419278	0.058644247
9	108	0.040700071	0.059781609	0.059094547
9	144	0.044477086	0.053391154	0.03202526
9	180	0.054863592	0.053195501	0.036634255
9	216	0.053230993	0.0661019	0.035833067
9	252	0.04226616	0.073033853	0.035603904
10	36	0.014607595	0.028280949	0.045866925
10	72	0.02816648	0.044627679	0.041308571
10	108	0.039530678	0.059659499	0.046008508
10	144	0.026959654	0.042268527	0.021480541
10	180	0.035403479	0.033531714	0.020972049

Continued on next page

Table A.4 – *Continued from previous page*

k	mbr	Latitude	Longitude	LBS
10	216	0.043485771	0.047432957	0.024991763
10	252	0.037342133	0.055789477	0.02634867
11	36	0.014940282	0.029444483	0.03387408
11	72	0.027840176	0.043017423	0.042520415
11	108	0.033730503	0.054157389	0.034607713
11	144	0.05182558	0.067260299	0.023651753
11	180	0.068500035	0.06989317	0.041058177
11	216	0.062987855	0.095494629	0.046124053
11	252	0.065988077	0.112996498	0.031087909
12	36	0.016656	0.038622199	0.023167287
12	72	0.031730663	0.010102357	0.025714212
12	108	0.025230026	0.046929822	0.025966526
12	144	0.062911205	0.074114572	0.047170123
12	180	0.020518691	0.127989845	0.024003235
12	216	0.045365055	0.068916761	0.016139183
12	252	0.071896305	0.079727567	0.025003549

Table A.5: Average Generalization Time

k	mbr	Processing (ms)	Wait (ms)	Turnaround (ms)
1	36	0.253107961	0	0.253107961
1	72	0.385436745	0	0.385436745
1	108	0.436895769	0	0.436895769
1	144	0.384693978	0	0.384693978
1	180	0.457208173	0	0.457208173
1	216	0.414558428	0	0.414558428
1	252	0.363879184	0	0.363879184
2	36	1.034204001	71.10724862	73.17565662
2	72	0.889716226	17.65871035	19.4381428
2	108	0.808591556	8.136275565	9.753458678
2	144	0.817483352	5.249483993	6.884450698
2	180	0.766131847	3.166394478	4.698658173
2	216	0.668815354	2.005116006	3.342746714
2	252	0.600261504	1.503890122	2.70441313
3	36	2.622752842	491.2808451	499.1491037
3	72	1.777575999	101.591578	106.924306
3	108	1.640686197	44.02348792	48.94554651
3	144	1.388802097	29.65557278	33.82197908
3	180	1.25650545	16.1323912	19.90190755
3	216	1.129552899	8.883940829	12.27259953
3	252	0.942624115	6.311566594	9.139438941
4	36	7.861772195	2620.642722	2652.089811

Continued on next page

Table A.5 – *Continued from previous page*

k	mbr	Processing (ms)	Wait (ms)	Turnaround (ms)
4	72	3.848735484	516.6523182	532.0472601
4	108	3.0873596	208.3178708	220.6673092
4	144	2.960779338	136.6210309	148.4641482
4	180	2.594377712	76.3349346	86.71244544
4	216	2.170910678	39.49565474	48.17929745
4	252	1.847057008	27.85817903	35.24640706
5	36	16.0008502	4975.206082	5055.210333
5	72	7.907684226	1277.363616	1316.902037
5	108	6.257659387	650.3863978	681.6746947
5	144	5.825741153	412.5298517	441.6585575
5	180	5.352661694	302.6469925	329.410301
5	216	3.803288726	240.6451942	259.6616378
5	252	3.44090965	165.691438	182.8959862
6	36	39.18439624	11154.63231	11389.73869
6	72	19.20963306	2784.659013	2899.916811
6	108	15.33742996	1309.012925	1401.037505
6	144	14.63355644	5876.529555	5964.330893
6	180	9.176394143	6799.997287	6855.055652
6	216	7.227602195	2913.260082	2956.625695
6	252	5.770132762	2749.907005	2784.527802
7	36	96.74495333	32772.83138	33450.04606
7	72	49.91560124	7349.667159	7699.076367
7	108	40.8062244	3455.635102	3741.278673
7	144	33.33202057	14442.40119	14675.72533
7	180	27.59577649	12782.33535	12975.50578
7	216	18.92090747	4463.778594	4596.224947
7	252	19.45608758	5422.949601	5559.142214
8	36	1679.925852	978787.1481	992226.5549
8	72	1075.761539	178784.0603	187390.1526
8	108	415.786874	51961.94497	55288.23996
8	144	353.4881488	52767.78538	55595.69057
8	180	175.6996125	30451.07756	31856.67446
8	216	172.4398466	14989.92252	16369.44129
8	252	109.9881292	12303.38375	13183.28878
9	36	4817.977849	1738493.364	1781855.165
9	72	2144.710324	294776.8116	314079.2045
9	108	1602.823674	98003.32918	112428.7422
9	144	1077.050626	139474.5241	149167.9797
9	180	669.4562534	87417.98251	93443.08879
9	216	572.4226274	47013.51342	52165.31707
9	252	317.8748272	33705.08005	36565.95349
10	36	18897.58767	3793792.293	3982768.17

Continued on next page

Table A.5 – *Continued from previous page*

k	mbr	Processing (ms)	Wait (ms)	Turnaround (ms)
10	72	10043.95016	636542.3666	736981.8682
10	108	4996.109091	198419.7745	248380.8655
10	144	3661.336735	506316.7507	542930.1181
10	180	2422.682343	339446.67	363673.4934
10	216	2026.002941	114146.6961	134406.7255
10	252	1087.738134	95157.38789	106034.7692
11	36	25365.55102	7763448.502	8042469.563
11	72	22049.70677	1350612.286	1593159.06
11	108	14481.07438	386818.3719	546110.1901
11	144	13103.8129	299888.5097	444030.4516
11	180	8890.023622	129689.3228	227479.5827
11	216	8025.814159	87438.47788	175722.4336
11	252	4897.716418	43525.90299	97400.78358
12	36	30238.75	24700343.67	25063208.67
12	72	36885.2	5420809.4	5863431.8
12	108	34520.33333	2439758.167	2854002.167
12	144	26495.25	2649958.625	2967901.625
12	180	18687.14286	744434.1429	968679.8571
12	216	23167.2	1146606.2	1424612.6
12	252	10038	455945.7143	576401.7143

Table A.6: Average STD of Generalization Time

k	mbr	Processing (ms)	Wait (ms)	Turnaround (ms)
1	36	1.96358292	0	1.96358292
1	72	2.463672966	0	2.463672966
1	108	6.96407595	0	6.96407595
1	144	2.421065747	0	2.421065747
1	180	7.108534799	0	7.108534799
1	216	7.268984025	0	7.268984025
1	252	2.406638721	0	2.406638721
2	36	3.719487347	106.436722	106.7482519
2	72	3.617670752	38.30682441	38.94572032
2	108	3.515517126	18.12312065	19.27781484
2	144	3.497862692	24.22019289	25.17625774
2	180	3.359863544	7.323607987	9.666249746
2	216	3.188972511	14.06001368	15.72182836
2	252	3.000066184	10.36110726	12.51691764
3	36	5.46219861	486.1441511	486.8002156
3	72	4.91253134	129.1824712	129.9679495
3	108	10.81349658	65.21637617	80.74220577
3	144	4.426726486	59.85512829	61.6190609

Continued on next page

Table A.6 – *Continued from previous page*

k	mbr	Processing (ms)	Wait (ms)	Turnaround (ms)
3	180	4.216029104	32.45819875	34.9785983
3	216	4.018292871	28.36477192	30.99786347
3	252	3.771688262	15.54458977	19.02016037
4	36	11.7555436	2327.472278	2332.448005
4	72	6.555696393	599.4091888	601.5902091
4	108	6.308084955	225.6348567	227.2933404
4	144	5.873725463	203.3397137	205.9155451
4	180	5.756364907	134.9288273	137.4534199
4	216	11.81120295	69.06033019	95.54402731
4	252	4.20078346	72.68672218	76.33237478
5	36	12.31947936	4315.015242	4324.069789
5	72	8.898819406	1079.287748	1086.781291
5	108	8.137525521	690.7118658	696.4748134
5	144	8.111839097	472.8618721	479.2947992
5	180	8.083011642	326.1629259	331.752818
5	216	6.302196817	327.2067377	333.594623
5	252	15.06562553	229.7100048	266.0554967
6	36	31.09592858	8984.498733	9009.984668
6	72	21.07827864	2262.475375	2286.991319
6	108	15.27196347	1171.855929	1194.29176
6	144	11.99042157	656.9965292	675.6293582
6	180	9.705807591	499.625851	519.853453
6	216	8.132141232	362.2641002	380.5120251
6	252	6.417586185	295.7670516	310.1288778
7	36	60.31352736	27180.85195	27243.94007
7	72	51.05633094	6101.114642	6183.678841
7	108	35.97479528	3146.282521	3225.490972
7	144	30.34159308	1361.950305	1438.86859
7	180	23.62277033	1069.752156	1142.724437
7	216	17.64579192	696.7556449	759.9480621
7	252	12.66920347	604.081029	646.98553
8	36	2520.530606	835532.1999	840801.0853
8	72	2312.331917	194882.4664	201478.2165
8	108	550.5228342	77121.66644	77440.79628
8	144	497.0280835	31968.31423	32575.5229
8	180	177.8764969	8664.327803	9172.366665
8	216	220.1135743	10028.40771	10757.50193
8	252	86.75150511	4429.54602	4667.988856
9	36	9633.880533	1912085.452	1933059.044
9	72	3699.209305	346174.2384	354408.1009
9	108	3025.122322	114117.0254	121594.4488
9	144	2093.647536	62072.29003	66146.8071

Continued on next page

Table A.6 – *Continued from previous page*

k	mbr	Processing (ms)	Wait (ms)	Turnaround (ms)
9	180	919.4456501	24488.50235	27501.70007
9	216	721.7572108	19921.40098	22771.01697
9	252	345.8375576	8202.144088	10080.12948
10	36	123923.0019	3863270.83	4102400.798
10	72	22467.55516	663791.1865	745115.1621
10	108	8522.548716	246403.4048	270368.713
10	144	9148.616417	99927.8812	145813.5876
10	180	3544.093001	36047.32814	54862.67139
10	216	3363.052701	31102.2559	55598.35468
10	252	1347.924471	13173.60608	23868.37198
11	36	33546.34673	7731496.702	7806682.297
11	72	44791.72897	1369108.128	1619202.015
11	108	24451.48312	282780.2884	454708.8244
11	144	17737.90622	281910.0593	375078.964
11	180	16105.68095	197803.2417	295600.0617
11	216	11673.35685	96380.48262	199055.6448
11	252	6197.944965	53055.36202	101682.5357
12	36	26437.16622	11979934.03	12084523
12	72	29831.88041	2993636.811	3276795.37
12	108	40987.3182	825480.8954	1134758.627
12	144	25538.98752	1507622.305	1527478.561
12	180	24749.616	481461.4804	759188.3068
12	216	22273.11256	642202.1161	880196.8399
12	252	5972.794584	273960.1507	285853.991