Quality Metrics for Optimizing Parameters Tuning in Clustering Algorithms for Extraction of Points of Interest in Human Mobility

Miguel Nuẽz del Prado Cortez
Peru I+D+I
Technopark IDI
miguel.nunez@peruidi.com

Hugo Alatrista-Salas
GRPIAA Labs., PUCP
Peru I+D+I
halatrista@pucp.pe

Abstract

Clustering is an unsupervised learning technique used to group a set of elements into non-overlapping clusters based on some predefined dissimilarity function. In our context, we rely on clustering algorithms to extract points of interest in human mobility as an inference attack for quantifying the impact of the privacy breach. Thus, we focus on the input parameters selection for the clustering algorithm, which is not a trivial task due to the direct impact of these parameters in the result of the attack. Namely, if we use too relax parameters we will have too many point of interest but if we use a too restrictive set of parameters, we will find too few groups. Accordingly, to solve this problem, we propose a method to select the best parameters to extract the optimal number of POIs based on quality metrics.

1 Introduction

The first step in inference attacks over mobility traces is the extraction of the point of interest (POI) from a trail of mobility traces. Indeed, this phase impacts directly the global accuracy of an inference attack that relies on POI extraction. For instance, if an adversary wants to discover Alice's home and place of work the result of the extraction must be as accurate as possible, otherwise they can confuse or just not find important places. In addition, for a more sophisticated attack such as next place prediction, a mistake when extracting POIs can decrease significantly the global precision of the inference. Most of the extraction techniques use heuristics and clustering algorithms to extract POIs from location data. On one hand, heuristics rely on the *dwell time*, which is the lost of signal when user gets into a building. Another used heuristic is the *residence time*, which represents the time that a user spends at a particular place. On the other hand, clustering algorithms group nearby mobility traces into clusters.

In particular, in the context of POI extraction, it is important to find a suitable set of parameters, for a specific cluster algorithm, in order to obtain a good accuracy as result of the clustering. The main contribution of this paper is a methodology to find a "optimal" configuration of input parameters for a clustering algorithm based on quality indices. This optimal set of parameters allows us to have the appropriate number of POIs in order to perform another inference attack. This paper is organized as follows. First, we present some related works on parameters estimation techniques in Section 2. Afterwards, we describe the clustering algorithms used to perform the extraction of points of interests (POIs) as well as the metrics to measure the quality of formed clusters in sections 3 and 4, respectively. Then, we introduce the method to optimize the choice of the parameters in Section 5. Finally, Section 6 summarizes the results and presents the future directions of this paper.

2 Related works

Most of the previous works estimate the parameters of the clustering algorithms for the point of interest extraction by using empirical approaches or highly computationally expensive methods. For instance, we use for illustration purpose two classical clustering approaches, K-means (MacQueen et al., 1967) and DBSCAN (Ester et al., 1996). In the former

clustering algorithm, the main issue is how to determine k, the number of clusters. Therefore, several approaches have been proposed to address this issue (Hamerly and Elkan, 2003; Pham et al., 2005). The latter algorithm relies on OPTICS (Ankerst et al., 1999) algorithm, which searches the space of parameters of DBSCAN in order to find the optimal number of clusters. The more parameters the clustering algorithm has, the bigger the combinatorial space of parameters is. Nevertheless, the methods to calibrate cluster algorithm inputs do not guarantee a good accuracy for extracting meaningful POIs. In the next section, we described the cluster algorithms used in our study.

3 Clustering algorithms for extraction of points of interest

To perform the POI extraction, we rely on the following clustering algorithms:

3.1 Density Joinable Cluster (DJ-Cluster)

DJ-Cluster (Zhou et al., 2004) is a clustering algorithm taking as input a minimal number of points $minpts$, a radius r and a trail of mobility traces M. This algorithm works in two phases. First, the pre-processing phase discards all the moving points (*i.e.* whose speed is above ϵ, for ϵ a small value) and then, squashes series of repeated static points into a single occurrence for each series. Next, the second phase clusters the remaining points based on neighborhood density. More precisely, the number of points in the neighborhood must be equal or greater than $minpts$ and these points must be within radius r from the medoid of a set of points. Where medioid is the real point *m* that minimizes the sum of distances from the point *m* to the other points in the cluster. Then, the algorithm merges the new cluster with the clusters already computed, which share at least one common point. Finally, during the merging, the algorithm erases old computed clusters and only keeps the new cluster, which contains all the other merged clusters.

3.2 Density Time Cluster (DT-Cluster)

DT-Cluster (Hariharan and Toyama, 2004) is an iterative clustering algorithm taking as input a distance threshold d, a time threshold t and a trail of mobility traces M. First, the algorithm starts by building a cluster C composed of all the consecutive points within distance d from each other. Afterwards, the algorithm checks if the accumulated time of mobility traces between the youngest and the oldest ones is greater than the threshold t. If it is the case, the cluster is created and added to the list of POIs. Finally as a post-processing step, DT-Cluster merges the clusters whose medioids are less than $d/3$ far from each other.

3.3 Time Density (TD-Cluster)

Introduced in (Gambs et al., 2011), TD-Cluster is a clustering algorithm inspired from DT Cluster, which takes as input parameters a radius r, a time window t, a tolerance rate τ, a distance threshold d and a trail of mobility traces M. The algorithm starts by building iteratively clusters from a trail M of mobility traces that are located within the time window t. Afterwards, for each cluster, if a fraction of the points (above the tolerance rate τ) are within radius r from the medoid, the cluster is integrated to the list of clusters outputted, whereas otherwise it is simply discarded. Finally, as for DT Cluster, the algorithm merges the clusters whose medoids are less than d far from each other.

3.4 Begin-end heuristic

The objective of the *begin and end location finder* inference attack (Gambs et al., 2010) is to take as meaningful points the first and last of a journey. More precisely, this heuristic considers that the beginning and ending locations of a user, for each working day, might convey some meaningful information.

Since we have introduced the different clustering algorithms to extract points of interest, we present in the next section the indices to measure the quality of the clusters.

4 Cluster quality indices

One aspect of the extraction of POIs inference attacks is the quality of the obtained clusters, which impacts on the precision and recall of the attack. In the following subsection we describe some metrics to quantify how accurate or "how good" is the outcome of the clustering task. Intuitively, a good clustering is one that identifies a group of clusters that are well separated one from each other, compact

and representative. Table 1 summarizes the notation used in this section.

Symbol	Definition
C	An ensemble of clusters.
c_i	The i^{th} cluster of C.
n_c	The number of clusters in C.
m_i	The medoid point of the i^{th} cluster.
$\mathsf{d}(x, y)$	The Euclidean distance between x and y.
$\|c_i\|$	The number of points in a cluster c_i.
m'	The closest point to the medoid m_i.
m''	The second closest point to the medoid m_i.
$\|C\|$	The total number of points in a set of C.

Table 1: Summary of notations

4.1 Intra-inter cluster ratio

The *intra-inter* cluster ratio (Hillenmeyer, 2012) measures the relation between compact (Equation 1) and well separated groups (Equation 3). More precisely, we first take the inter-cluster distance, which is the average distance from each point in a cluster c_i to its medoid m_i.

$$DIC(c_i) = \frac{1}{|c_i| - 1} \sum_{x_j \in c_i, x_j \neq m_i}^{|c_i|} d(x_j, m_i) \quad (1)$$

Then, the average intra-cluster distance (***DIC***) is computed using Equation 2.

$$AVG\_DIC(C) = \frac{1}{n_c} \sum_{c_i \in C}^{|C|} DIC(c_i) \quad (2)$$

Afterwards, the mean distance among all medoids (***DOC***) in the cluster C is computed, using Equation 3.

$$DOC(C) = \frac{1}{|n_C| \times (|n_C| - 1)} \sum_{c_i \in C}^{|C|} \sum_{c_j \in C, i \neq j}^{|C|} d(m_i, m_j) \quad (3)$$

Finally, the ratio intra-inter cluster *rii* is given by the Equation 4 as the relationship between the average intra cluster distance divided by the inter-cluster distance.

$$rii(C) = \frac{AVG\_DIC(C)}{DOC(C)} \quad (4)$$

The intra-inter ratio has an approximate linear complexity in the number of points to be computed and gives low values to well separated and compact cluster.

4.2 Additive margin

Inspired by the Ben-David and Ackerman (Ben-David and Ackerman, 2008) *k-additive Point Margin* (*K-AM*) metric , which evaluates how well centered clusters are. We measure the difference between the medoid m_i and its two closest points m' and m'' of a given group c_i belonging to a cluster C (Equation 5).

$$K - AM(c_i) = d(m_i, m_i'') - d(m_i, m_i') \quad (5)$$

Since the average of the k-additive point margins for all groups c_i in a cluster C is computed, we take the ratio between the average k-additive Point Margin and the minimal inter-cluster distance (Equation 1) as shown in Equation 6.

$$AM(C) = min_{c_i \in C} \frac{\frac{1}{n_c} \sum_{c_i \in C}^{n_c} K - AM(c_i)}{DIC(c_i)} \quad (6)$$

The additive margin method has a linear complexity in the number of clusters. This metric gives a high value for a well centered clusters.

4.3 Information loss

The information loss ratio is a metric inspired by the work of Sole and coauthors (Solé et al., 2012). The basic idea is to measure the percent of information that is lost while representing original data only by a certain number of groups (*e.g.*, when we represent the POIs by the cluster medoids instead of the whole set of points). To evaluate the percent of information loss, we compute the sum of distance of each point represented by x_i to its medoid m_i for all clusters $c_i \in C$ as we shown in Equation 7.

$$SSE(C) = \sum_{c_i \in C}^{n_c} \sum_{x_j \in c_i}^{|c|} d(x_j, m_i) \quad (7)$$

Then, we estimate the accumulated distance of all points of a trail of mobility traces in the cluster C to a global centroid (GC) using the following equation Equation 8.

$$SST(C) = \sum_{x_i \in C}^{|C|} d(x_i, GC) \quad (8)$$

Finally, the ratio between aforementioned distances is computed using Equation 9, which results in the

information loss ratio.

$$IL(C) = \frac{SSE(C)}{SST(C)} \quad (9)$$

The computation of this ratio has a linear complexity. The lowest is the value of this ratio, the more representative the clusters are.

4.4 Dunn index

This quality index (Dunn, 1973; Halkidi et al., 2001) attempts to recognize compact and well-separated clusters. The computation of this index relies on a dissimilarity function (*e.g.* Euclidean distance d) between medoids and the diameter of a cluster (*c.f*, Equation 10) as a measure of dispersion.

$$diam(c_i) = max_{x,y \in c_i, x \neq y} d(x, y) \quad (10)$$

Then, if the clustering C is compact (*i.e*, the diameters tend to be small) and well separated (distance between cluster medoids are large), the result of the index, given by the Equation 11, is expected to be large.

$$DIL(C) = min_{c_i \in C}[min_{c_j \in C, j=i+1} [\frac{d(m_i, m_j)}{max_{c_k \in C} diam(c_k)}]] \quad (11)$$

The greater is this index, the better the performance of the clustering algorithm is assumed to be. The main drawbacks of this index is the computational complexity and the sensitivity to noise in data.

4.5 Davis-Bouldin index

The objective of the Davis-Bouldin index (*DBI*) (Davies and Bouldin, 1979; Halkidi et al., 2001) is to evaluate how well the clustering was performed by using properties inherent to the dataset considered. First, we use a scatter function within the cluster c_i of the clustering C (Equation 12).

$$S(c_i) = \sqrt{\frac{1}{n_c} \sum_{x_j \in c_i}^{n} d(x_j, m_i)^2} \quad (12)$$

Then, we compute the distance between two different clusters c_i and c_j, given by Equation 13.

$$M(c_i, c_j) = \sqrt{d(m_i, m_j)} \quad (13)$$

Afterwards, a similarity measure between two clusters c_i and c_j, called *R*-similarity, is estimated, based on Equation 14.

$$R(c_i, c_j) = \frac{S(c_i) + S(c_j)}{M(c_i, c_j)} \quad (14)$$

After that, the most similar cluster c_j to c_i is the one maximizing the result of the function $R_{all}(c_i)$, which is given by Equation 15 for $i \neq j$.

$$R_{all}(c_i) = max_{c_j \in C, i \neq j} R(c_i, c_j) \quad (15)$$

Finally, the DBI is equal to the average of the similarity between clusters in the clustering set C (Equation 16).

$$DBI(C) = \frac{1}{n_c} \sum_{c_i \in C}^{n_c} R_{all}(c_i) \quad (16)$$

Ideally, the clusters $c_i \in C$ should have the minimum possible similarity to each other. Accordingly, the lower is the DB index, the better is the clustering formed. These indices would be used to maximize the number of significant places a cluster algorithm could find. More precisely, in the next section we evaluate the cluster algorithm aforementioned as well as the method to extract the meaningful places using the quality indices.

5 Selecting the optimal parameters for clustering

In order to establish how to select the best set of parameters for a given clustering algorithm, we have computed the precision, recall and F-measure of all users of LifeMap dataset (Chon and Cha, 2011). One of the unique characteristic of this dataset is that the POIs have been annotated by the users. Consequently, given a set of clusters $c_i \in C$ such that $C = \{c_1, c_2, c_3, \ldots, c_n\}$ and a set of points of interest (POIs) defined by the users $P_{poi} = \{p_{poi\ 1}, p_{poi\ 2}, p_{poi\ 3}, \ldots, p_{poi\ n}\}$ we were able to compute the precision, recall and f-measure as we detail in the next subsection.

5.1 Precision, recall and F-measure

To compute the recall (*c.f.* Equation 17), we take as input a clustering set C, the *ground truth* represented by the vector P_{poi} (which was defined manually by

each user) as well as a $radius$ to count all the clusters $c \in C$ that are within the $radius$ of $p_{poi} \in P_{poi}$, which represents the "good clusters". Then, the ratio of the number of *good clusters* compared to the *total number of found clusters* is computed. This measure illustrates the ratio of extracted cluster that are POIs divided by the total number of extracted clusters.

$$Precision = \frac{good\ clusters}{total\ number\ extracted\ clusters} \quad (17)$$

To compute the recall (*c.f.* Equation 18), we take as input a clustering set C, a vector of POIs P_{poi} as well as a $radius$ to count the discovered POIs $p_{poi} \in P_{poi}$ within a $radius$ of the clusters $c \in C$, which represents the "good POIs". Then, the ratio between the number of *good POIs* and the *total number of POIs* is evaluated. This metric represents the percent of the extracted unique POIs.

$$Recall = \frac{good\ POIs}{total\ number\ of\ POIs} \quad (18)$$

Finally, the F-measure is defined as the weighted average of the precision and recall as we can see in Equation 19.

$$F - measure = 2 \times \frac{precision \times recall}{precision + recall} \quad (19)$$

We present the dataset used for our experiments in the next subsection.

5.2 Dataset description

In order to evaluate our approach, we use the *LifeMap dataset* (Chon et al., 2012), which is composed of mobility traces of 12 user collected for a year in Seoul, Korea. This dataset comprises location (latitude and longitude) collected with a frequency between 2 to 5 minutes with the user defined point of interest as *true* if the mobility trace is considered as important or meaningfull for each user. Table 2 summarizes the main characteristics of this dataset, such as the collect period, the average number of traces per user, the total number of mobility traces in the dataset, the minimal and maximal number of users' mobility traces.

Since we have described our dataset, we present the results of our experiments in the next subsection.

Characteristics	LifeMap
Total nb of users	12
Collection period (nb of days)	366
Average nb of traces/user	4 224
Total nb of traces	50 685
Min #Traces for a user	307
Max #Traces for a user	9 473

Table 2: Main characteristics of the LifeMap dataset.

5.3 Experimental results

This section is composed of two parts, in the first part we compare the performance of the previously described clustering algorithms, with two baseline clustering algorithms namely *k*-means and DBSCAN. In the second part, a method to select the most suitable parameters for a clustering algorithm is presented.

Input parameters	Possible values	DBSCAN	DJ cluster	DT cluster	K-means	TD cluster
Tolerance rate (%)	{0.75, 0.8, 0.85, 0.9}	y	Y	y	y	y
Tolerance rate (%)	{0.75, 0.8, 0.85, 0.9}	✗	✗	✗	✗	✓
Minpts (points)	{3, 4, 5, 6, 7, 8, 9, 10, 20, 50}	✓	✓	✗	✗	✗
Eps (Km.)	{0.01, 0.02, 0.05, 0.1, 0.2}	✓	✓	✓	✗	✓
Merge distance (Km.)	{0.02, 0.04, 0.1, 0.2, 0.4}	✗	✗	✓	✗	✓
Time shift (hour)	{1, 2, 3, 4, 5, 6}	✗	✗	✓	✗	✓
K (num. clusters)	{5, 6, 7,8, 9}	✗	✗	✗	✓	✗

Table 3: Summary of input parameters for clustering algorithms.

	Precision	Recall	F-measure	Time(s)	Number of parameters	Complexity
DBSCAN	0,58	0,54	0,48	316	3	$O(n^2)$
DJ-Cluster	0,74	0,52	0,52	429	3	$O(n^2)$
DT-Cluster	0,38	0,47	0,39	279	3	$O(n^2)$
k-means	0,58	0,51	0,49	299	2	$O(n)$
TD-Cluster	0,43	0,54	0,44	362	4	$O(n^2)$

Table 4: The characteristics of the clustering algorithms.

In order to compare the aforementioned clustering algorithms, we have take into account the precision, recall, F-measure obtained, average execution time, number of input parameters and time complexity. To evaluate these algorithms, we used the LifeMap dataset with POIs annotation and a set of different parameters configurations for each algorithm, which are summarized in Table 3. After running these con-

figurations, we obtained the results shown in Table 4 for the different input values.

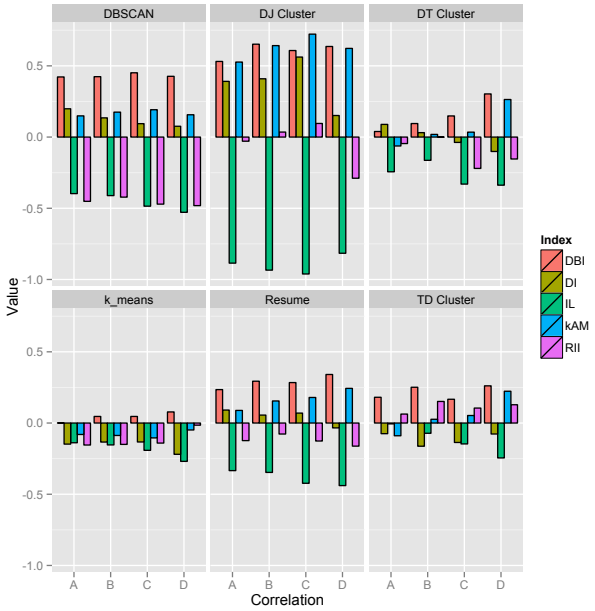

Figure 1: Correlation of quality indices with the computed F-measure. Where A) is the correlation measured between the user annotation and the centroid at 20 m of radius B) at 35 m radius C) at 50 m radius, D) at 100 m radius and DBI=Davis-Bouldin index, DI=Dunn index, IL=Information loss, kAM=Additive margin and RII= Ratio intra-inter cluster.

It is possible to observe that the precision of DJ-Cluster out performs better than the other clustering algorithms. In terms of recall DBSCAN and TD-Cluster perform the best but DJ-Cluster is just behind them. Moreover, DJ-Cluster has the best F-measure. Regarding the execution time, DT-Clustering the fastest one while DJ-Cluster is the slowest algorithm due to the preprocessing phase. Despite the high computational time of DJ-Cluster, this algorithm performs well in terms of F-measure.

In the following, we describe our method to choose "optimal" parameters for obtaining a good F-measure. We have used the aforementioned algorithms with a different set of input parameters configurations for users with POIs annotations in the LifeMap dataset (Chon and Cha, 2011). Once clusters are built, we evaluate the clusters issued from different configurations of distinct algorithms using the previously described quality indices. Afterwards, we were able to estimate the precision, recall and F-measure using the manual annotation of POIs by the users in the LifeMap dataset.

Regarding the relation between the quality indices and the F-measure, we studied the relationship between these factors, in order to identify the indices that are highly correlated with the F-measure, as can be observed in Figure 1. We observe that the two best performing indices, except for k-means, are IL and DBI. The former shows a negative correlation with respect to the F-measure. While the latter, has a positive dependency to the F-measure. Our main objective is to be able to identify the relationship between quality and F-measure among the previous evaluated clustering algorithms. Accordingly, we discard the inter-intra cluster ratio (RII) and the adaptive margin (AM), which only perform well when using k-means and the DJ clustering algorithms. Finally, we observe that the Dunn index has a poor performance. Based on these observations, we were able to propose an algorithm to automatically choose the best configuration of input parameters.

5.4 Parameter selection method

Let us define a vector of parameters $p_i \in P$ and P a set of vectors, such that $P = \{p_1, p_2, p_3, \ldots, p_n\}$, a trail of mobility traces M of a user. From previous sections we have the clustering function $C(p_i)$ and the quality metrics Information Loss $IL(C)$ and Davis-Bouldin index $DBI(C)$. Thus, for each vector of parameters we have a tuple composed of the trail of mobility traces, the result of the clustering algorithm and the quality metrics $(p_i, M, C_{p_i}, IL_{C_{p_i}}, DBI_{C_{p_i}})$. When we compute the clustering algorithm and the quality metrics for each vector of parameter for a given user u. We define also a χ'_u matrix, which the matrix χ_u sorted by IL ascending. Finally, the result matrix χ_u is of the form:

$$\chi_u = \begin{vmatrix} p_1 & M & C_{p_1} & IL_{C_{p_1}} & DBI_{C_{p_1}} \\ p_2 & M & C_{p_2} & IL_{C_{p_2}} & DBI_{C_{p_2}} \\ p_3 & M & C_{p_3} & IL_{C_{p_3}} & DBI_{C_{p_3}} \\ \ldots & & & & \\ p_n & M & C_{p_n} & IL_{C_{pn}} & DBI_{C_{pn}} \end{vmatrix}$$

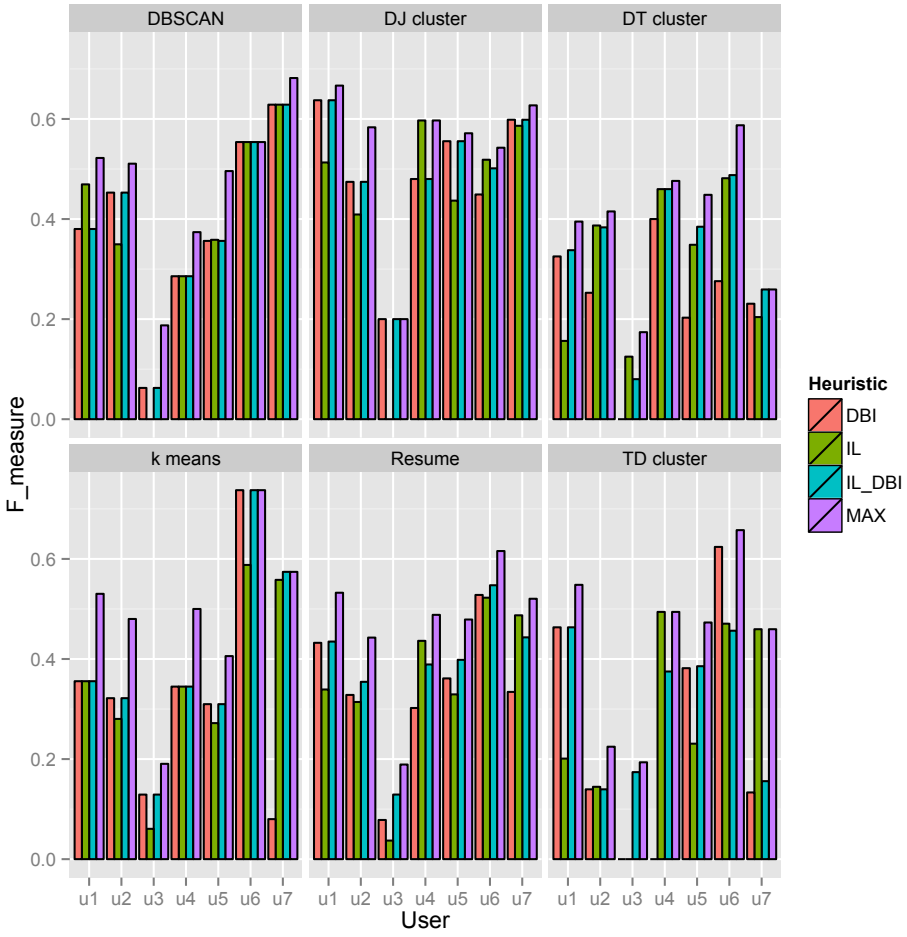

Figure 2: Comparison between F-measure and parameters selection based on schema in Figure **??**. Where DBI=Davis-Bouldin index, IL=Information loss, IL_DBI= combination of IL and DBI and MAX is the maximal computed F-measure (taken as reference to compare with IL_DBI). Resume is the average of all the results using different clustering algorithms.

Therefore, the parameter selection function $S(\chi_u)$ could be defined as:

$$S(\chi_u) = \begin{cases} p_i, & \text{if } max_{p_i}(DBI) \ \& \ min_{p_i}(IL) \\ p'_i, & \text{if } max_{p'_i}(DBI) \text{ in 1st quartile} \end{cases} \quad (20)$$

In detail, the function S takes as input a χ matrix containing the parameters vector p_i, a trail of mobility traces M, the computed clustering $C(p_i, M)$ as well as the quality metrics, such as Information loss ($IL(C)$) and the Davis-Bouldin index ($DBI(C)$). Once all these values have been computed for each evaluated set of parameters, two cases are possible. In the first case, both IL and DBI agree on the same set of input parameters. In the second situation, both IL and DBI refer each one to a different set of parameters. In this case, the algorithm sorts the values by IL in the ascending order (*i.e.*, from the smallest to the largest information loss value). Then, it chooses the set of parameters with the greatest DBI in the first quartile.

For the sake of evaluation, our methodology was tested using the LifeMap dataset to check if the chosen parameters are optimal. We have tested the method with the seven users of LifeMap that have annotated manually their POIs. Consequently, for every set of settings of each clustering algorithm, we have computed the F-measure because we have the

ground truth as depicted in Figure 2. The "MAX" bar represents the best F-measure for the given user and it is compare to the F-measures obtained when using the "DBI", "IL" or "IL_DBI" as indicators to choose the best input parameters configuration. Finally, this method has a satisfactory performance extracting a good number of POIs for maximizing the F-measure achieving a difference of only 9% with respect to the F-measure computed from the data with the ground truth.

6 Conclusion

In the current paper, we have presented a method to extract the a optimal number of POIs. Consequently, based on the method described in tis paper, we are able to find an appropriate number of POIs relying only on the quality metrics of the extracted clusters and without the knowledge of the ground truth. Nonetheless, we are aware of the small size of dataset but the results encourage us to continue in this direction.

Therefore, in the future we plan to test our method in a larger dataset and in presence of noise like downsamplig or random distortion. Another idea is to evaluate the impact of this method in more complex attacks like prediction of future locations or deanonymization to verify if this step can affect the global result of a chain of inference attacks.

References

Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. (1999). Optics: ordering points to identify the clustering structure. *ACM SIGMOD Record*, 28(2):49–60.

Ben-David, S. and Ackerman, M. (2008). Measures of clustering quality: A working set of axioms for clustering. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, pages 121–128, Vancouver, Canada.

Chon, Y. and Cha, H. (2011). LifeMap: A smartphone-based context provider for location-based services. *Pervasive Computing, IEEE*, 10(2):58–67.

Chon, Y., Talipov, E., Shin, H., and Cha, H. (2012). CRAWDAD data set yonsei/lifemap (v. 2012-01-03). Downloaded from http://crawdad.cs.dartmouth.edu/yonsei/lifemap.

Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.

Dunn, J. C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57.

Ester, M., peter Kriegel, H., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Knowledge Discovery and Data Mining*, 2(4):226–231.

Gambs, S., Killijian, M.-O., and Núñez del Prado Cortez, M. (2010). GEPETO: A GEoPrivacy-Enhancing TOolkit. In *Advanced Information Networking and Applications Workshops*, pages 1071–1076, Perth, Australia.

Gambs, S., Killijian, M.-O., and Núñez del Prado Cortez, M. (2011). Show me how you move and I will tell you who you are. *Transactions on Data Privacy*, 2(4):103–126.

Halkidi, M., Batistakis, Y., and Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(3):107–145.

Hamerly, G. and Elkan, C. (2003). Learning the k in K-means. In *In Neural Information Processing Systems*, pages 1–8, Vancouver, Canada.

Hariharan, R. and Toyama, K. (2004). Project lachesis: Parsing and modeling location histories. *Lecture notes in computer science - Geographic information science*, 3(1):106–124.

Hillenmeyer, M. (2012). Intra and inter cluster distance. `http://www.stanford.edu/~maureenh/quals/html/ml/node82.html`.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate mbservations. In *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, Berkeley, CA, USA.

Pham, D. T., Dimov, S. S., and Nguyen, C. D. (2005). Selection of K in K-means clustering. *Journal of Mechanical Engineering Science*, 205(1):103–119.

Solé, M., Muntés-Mulero, V., and Nin, J. (2012). Efficient microaggregation techniques for large numerical data volumes. *International Journal of Information Security - Special Issue: Supervisory control and data acquisition*, 11(4):253–267.

Zhou, C., Frankowski, D., Ludford, P., Shekhar, S., and Terveen, L. (2004). Discovering personal gazetteers: an interactive clustering approach. In *Proceedings of the annual ACM international workshop on Geographic information systems*, pages 266–273, New York, NY, USA.