A Study on Parameter Estimation for a Mining Flock Algorithm

Rebecca Ong[1], Mirco Nanni[1], Chiara Renso[1], Monica Wachowicz[3], and Dino Pedreschi[2]

[1] KDDLab, ISTI-CNR, Pisa,Italy
[2] KDDLab, University of Pisa, Italy
[3] University of New Brunswick, Fredericton, Canada

Abstract. Due to the diffusion of location-aware devices and location-based services, it is now possible to analyse the digital trajectories of human mobility through the use of mining algorithms. However, in most cases, these algorithms come with little support for the analyst to actually use them in real world applications. In particular, means for understanding how to choose the proper parameters are missing. This work improves the state-of-the-art of mobility data analysis by providing an experimental study on the use of data-driven parameter estimation measures for mining flock patterns. Experiments were conducted on two real world datasets, one dealing with pedestrian movements in a recreational park and the other with car movements in a coastal area. The study has shown promising results for estimating suitable values for parameters for flock patterns envisaging a formal framework for parameter evaluation in the near future, since the advent of more complex pattern algorithms will require the use of a larger number of parameters.

1 Introduction

The increasing availability of data pertaining to the movements of people and vehicles, such as GPS trajectories and mobile phone call records, has fostered in recent years a large body of research on the analysis and mining of these data, to the purpose of discovering the patterns and models of human mobility. Examples along this line include [2, 3], which highlight the broad diversity of mobility patterns. A few authors concentrated on the problem of characterising and detecting *flocks*, i.e., patterns describing a set of objects that stay closely together during an interval of time, either moving together along the same route (a *moving flock*), or staying together in a specific location (a *stationary flock*) [6, 4, 7].

Flocking patterns highlight groups with synchronised movements that stay together for a while and disappear afterwards. In this paper, we follow the definition where a flock is a group of at least k objects that, observed during a time interval ΔT with a sampling rate R, remain spatially close to each other within a distance ϵ. While this definition and other variations in literature are useful for detecting flocks, it is apparent that setting such parameters $(k, \epsilon, \Delta T, R)$ makes it complex for an analyst to use flock mining in different contexts. These parameters clearly depend on the data under analysis, and may be vary greatly in different settings. We observed remarkable differences in datasets pertaining to pedestrian and car movements, which are the two trajectory datasets used in this paper. Such differences can be expected as well when observing other types of moving objects, e.g., bird trajectories. Despite this complexity, no prior work addressed the problem of parameter setting, which is a barrier especially for mobility experts who would like to use flock mining as a black box. To this aim, we address the *parameter estimation* problem of finding appropriate values for the parameters using a systematic data-driven method, based on the trajectory dataset that is being analysed. This paper provides an empirical evaluation of the effects of parameters in two different moving objects datasets. This is an initial step towards delineating a data-driven parameter estimation method for flock mining.

The structure of this paper is as follows: Section 2 presents some related approaches in the literature. Section 3 provides a summary of the flock algorithm considered in this study while Section 4 describes the experiments performed on two datasets in order to study the effect of the different parameters. Finally, Section 5 sums up the conclusions derived from the study.

2 Related works

Although the problem of finding realistic parameter values in data mining is well recognized in literature, very few papers have addressed this problem. Paper [1] is a well known work that proposes a solution for parameter estimation, and has inspired our approach. In this work, the authors propose a heuristic technique for determining the parameter values of the density-based clustering algorithm DBSCAN: *Eps* (radius) and $MinCard$ (minimum cardinality of clusters). A function called *k-distance* is defined to compute the distance between each object and its k-th nearest neighbor. These values are then plotted with objects ordered in descending order, in the so-called *sorted k-distance plot*. This plot is then used to find an estimation of the parameters: given an object p, the k-distance(p) is the value of *Eps* while $MinCard$ is set to $k+1$. This setting means that all objects with an equal or smaller k-distance are the "core" objects. The threshold object which maximize the k-distance in the least dense cluster gives the desired values. This object can be found visually in the plot by identifying the first "valley". The objects plotted to the left of the threshold will be considered as noise while other objects will be assigned to some cluster.

Another related work on parameter estimation related to flocks is found in [5] where the authors propose a set of algorithms for detecting convoys in trajectory datasets. They proposed a guideline for determining the parameters δ and λ of the proposed Douglas-Peucker (DP) algorithm, with the purpose of optimizing the execution time. The optimal convoy algorithm is run on a pre-processed dataset where trajectories have been simplified using the DP algorithm, which uses δ as a tolerance value for constraining the spatial distance between the original and the simplified points. The algorithm uses another additional parameter λ, which refers to the length of the time partitions. The determination of a good value for δ has the goal of finding a trade-off value giving a good simplification of original trajectories while mantaining a tight enough distance. For finding a good value for δ authors propose to run the DP algorithm with δ set to 0. They consider the actual tolerance values at each simplification step and find the values with the largest difference with their neighbour before averaging them to obtain the final parameter value. Meanwhile, a good λ is computed by taking the average probability of each object having an intermediate simplified point that is not found in other trajectories. However, the parameter estimation techniques were applied to the preprocessing step of the convoy algorithm rather than applying it directly to the parameters related to the flock or convoy definition.

3 A Moving Flock Extraction Algorithm

The study for parameter estimation has been designed with reference to the flock algorithm introduced in [7]. The algorithm finds moving flocks, each of which is a group of objects consisting of at least *min_points* members that are spatially close together while moving from one location to another over a minimum time duration *min_time_slices*. The algorithm requires four user-defined parameters: *synchronisation_rate*(R) - refers to the rate, specified in seconds, at which observation points (e.g., GPS recordings) are sampled for each moving object; *min_time_slices*(ΔT) - is the minimum number of consecutive times slices for which the objects remain spatially close; *min_points*(κ) - is the minimum number of objects in a moving flock; *radius*(ϵ) - defines the spatial closeness of a group of moving objects at a specific time instance.

The following is a pseudocode of the moving flock algorithm:

Algorithm 1: Moving Flock($\mathcal{D}$, R, ΔT, κ, ϵ)

```
1  synchDataset = synchronise(𝒟, R));
2  for each traj in synchDataset
3    if traj is NOT marked
4      for each curr_point, a sampled point of current traj
5        1-flocks = computeSpatialNeighbour(curr_point, ε, synchDataset);
6      n-flocks = merge_adj_cand(1-flocks);
7      for each cand, a candidate flock in n-flocks
8        if count_time_slices(cand)>=ΔT and compute_extent(cand)>=ε
9          for each traj', a member of cand
10             mark traj';
11           F = F ∪ cand;
```

The algorithm initially samples the observation points in the input dataset at a regular interval R (line 1). The next steps are performed for each of the trajectories. The current unmarked trajectory is considered as the base member and for each of its time instances, points belonging to other trajectories are considered as neighbours if their x,y components are within ϵ distance to that of the base (lines 4-5), producing candidate flocks lasting for only 1 time slice (i.e., $1{-}flocks$). The $1{-}flocks$ with adjacent time slices and have at least κ members in common are then merged, producing $2{-}flocks$ candidates. Merging is recursively applied until $n{-}flocks$, which refer to the longest duration candidate flocks, are found (line 6). $n{-}flocks$ lasting for at least ΔT time slices and covering a spatial extent of at least ϵ are considered as moving flocks and their member trajectories are marked in order to reduce the number of base trajectories that need to be processed (lines 8-11).

4 Parameter Estimation

This section focuses on the investigation of the effects of individual parameters on the flock results to understand how suitable values can be selected. This study for parameter estimation has been designed with reference to the flock algorithm introduced in [7], to find moving flock patterns using the parameters k, ϵ, ΔT, and R as discussed in section 3.

We start with a description of the datasets used for the experiments and an overview of some flock quality measures since these are necessary to understand the impact of the parameters on the results. These are followed by a discussion of the effect of each parameter before closing with an approach on finding a suitable *radius* value.

4.1 Context Awareness and Flock Cohesion Distance

We performed the study on two datasets that have two entirely different settings of two different types of moving objects. The first dataset, called DNP, contains 370 trajectories, one for each visitor and consisting of a total of 141,826 sample points. These were recorded using GPS devices given to the visitors at the parking lots where they have started their visits. Due to the sparsity of this dataset, we have combined the data in different days into one day. The second dataset, called OctoPisa, contains the trajectories of $\approx$ 40,000 cars for a total of $\approx$ 1,500,000 travels covering a temporal span of 5 weeks in a coastal area of Tuscany around the city of Pisa. From this large dataset, we concentrated on a subset of trajectories occurring on June, 29, 2010 in order to be able to perform a more detailed study on a specific time period. This is one of the days with the highest number of moving cars. It contains 28,032 trajectories (corresponding to 557,201 observed GPS points) of 10,566 vehicles. The flock algorithm can be applied as well to the other days and the obtained flocks can be combined in a straightforward manner to obtain the flocks inherent in the entire dataset.

The initial set of parameter values that we used for the DNP dataset is as follows: $R = 5mins.$, $\Delta T = 3$, $\kappa = 3$, and $\epsilon = 150m$. Meanwhile, the initial set used for the OctoPisa dataset is as follows: $R = 1min.$, $\Delta T = 3$, $\kappa = 3$, and $\epsilon = 150m$. For both datasets, we maximized R to a value that does not cause large distortion (from the domain expert's perspective) among the input trajectories. We selected a value of 3 for both ΔT and κ since using 2 is too small while 4 is quite large for finding a good number of flocks. Then, using the values for R, ΔT, κ, and the type of entity (i.e., pedestrian and car) in consideration, we derived a feasible and logical value for ϵ. In observing the effects of the individual parameters, we only modify the value of the parameter in consideration and retain the initial values for the rest.

A first step in parameter estimation is to understand how the parameters influence the results obtained by the flock extraction algorithm. In doing so, it is important to have an objective measure of this influence in order to understand whether decreasing or increasing the parameter values improves or worsens the quality of discovered flocks.

In our study, we used three measures, which are extensions of measures used for cluster evaluation. These measures include cohesion, separation and silhouette coefficient.

Flock cohesion distance is a measure of spatial closeness among members of a discovered flock. It is analogous to the cohesion measure used for evaluating clusters but specifically applied to flock patterns. It can be computed using Equation 1, which evaluates a specific flock F_i by computing the distance between each flock member m_j with the base $m_{i(b)}$. Recall from Algorithm 1 that candidate flocks are found using each trajectory as a base. Each discovered flock F_i has $m_{i(b)}$ as its base member. Members of F_i are spatially close to $m_{i(b)}$ for its duration of flocking. $|F_i|$ is the number of F_i's members.

$$flock\ coh(F_i) = \frac{\sum_{\substack{m_j \in F_i \\ m_j \neq m_{i(b)}}} prox_{intra}(m_j, m_{i(b)})}{|F_i| - 1} \tag{1}$$

The $prox_{intra}$ between a flock member m_j and the flock base $m_{i(b)}$ can be computed by averaging the Euclidean distance among (x, y) points that were sampled simultaneously as described in Equation 2. T refers to the flocking duration and it consists of a set of sampled time instances. x_j^t and y_j^t refer to the x, y components of member j at time instance t. $x_{i(b)}^t$ and $y_{i(b)}^t$ are the x and y components of flock i's base.

$$prox_{intra}(m_j, m_{i(b)}) = \frac{\sum_{t \in T} euclDist((x_j^t, y_j^t), (x_{i(b)}^t, y_{i(b)}^t))}{|T|} \tag{2}$$

The overall flock cohesion distance of an obtained flock result can be computed by averaging the flock cohesion distance scores for each flock in the result as shown in Equation 3. Naturally, a flock with a low cohesion distance score is considered as a high quality flock.

$$overall\ flock\ cohesion\ distance(F) = \frac{\sum_{F_i \in F} flock\ coh(F_i)}{|F|} \tag{3}$$

On the contrary, flock separation is a measure of spatial or spatio-temporal detachment of a flock from the rest and it can be computed using Equation 4.

$$flock\ sep(F_i) = \sum_{\substack{F_j \in F \\ i \neq j}} prox_{inter}(m_{b(i)}, m_{b(j)}) \tag{4}$$

While $prox_{intra}$ measures the distance among members of a flock, $prox_{inter}$ measures the distance among different flocks. The distance between a pair of flocks is computed by computing the distance between their respective bases. We propose two approaches for this computation: $prox_{inter(XYT)}$ and $prox_{inter(routeSim)}$. $prox_{inter(XYT)}$ considers the spatio-temporal components in computing the distance while $prox_{inter(routeSim)}$ only considers the spatial components. Using $prox_{inter(routeSim)}$, distance is computed based on the similarity between the route followed by the flocks, without considering co-occurrence of the route similarity in time. Choosing between these two depends on the similarity level that the user is interested in.

$prox_{inter(XYT)}$ computes the spatial distance among the portion of the base trajectories that overlap in time as was done for $prox_{intra}$. The remaining portion that does not overlap incurs a penalty *pnlty*, which is the maximum possible distance obtained from the overlapping portion plus an arbitrary value. In the case that this maximum value does not exist since the bases being compared are disjoint, a maximum penalty score is incurred. More specifically, Equation 5 describes how $prox_{inter(XYT)}$ is computed. $nonOverlapLTI$ refers to the number of un-matched time instances in the longer trajectory, *euclDistOvlp* is the sum of Euclidean distances among pairs of points (from each base) that overlap in time, and $maxTD$ refers to the length of the longer base trajectory in terms of the number of time instances.

$$prox_{inter(XYT)} = \frac{pnlty * (nonOverLTI) + euclDistOvlp}{maxTD * (|F_i| - 1)} \tag{5}$$

For $prox_{inter(routeSim)}$, we adapted an existing algorithm for computing the route similarity distance. The algorithm ignores the temporal component of the bases and computes the distance in terms of the spatial components by comparing the shape of the trajectories.

As with the overall flock cohesion of an obtained flock result, the overall flock separation can be computed by averaging individual flock separation scores.

Finally, the flock silhouette coefficient is a combination of the previously discussed measures as shown in Equation 6. Note that computed scores can range from -1 (large intraflock distances and small interflock distances) to 1 (small intraflock distances and large interflock distances).

$$flock\ Sil(F_i) = \frac{flock\ sep(F_i) - flock\ coh(F_i)}{max\{flock\ sep(F_i), flock\ coh(F_i)\}} \tag{6}$$

As with overall flock cohesion and separation, the overall silhouette coefficient of a flock result can be computed by the averaging silhouette score of each flock.

4.2 Observing the Effect of Varying the Parameters

This part discusses the observations derived from investigating the effect of different parameter values on the obtained flock results for the DNP and the OctoPisa datasets. The following subsections provides a discussion of the individual effect of each parameter.

Effect of *synchronisation_rate* (*R*) Out of the 4 parameters of the algorithm, the *synchronisation_rate* can affect the quality of the input dataset. More specifically, a very large value of R can distort the input trajectories whereas a very small value requires a longer processing time.

We have observed how dataset cohesion changes for different values of R. Dataset cohesion describes how each individual trajectory is cohesive with respect to the rest of trajectories in the dataset. To compute this value, we applied the flock separation measure and treated each trajectory as a base trajectory.

Experiments demonstrate that the synchronisation step can indeed modify the input and its cohesiveness but at the same time, the variation is small in the two datasets for smaller values of R. Using XYT cohesion, the largest difference between the smallest cohesion score compared to the other scores obtained using larger R is 371.67m when $R = 11mins.$ in the DNP dataset. Meanwhile, the largest difference is 1987.77m using route similarity cohesion when $R = 15mins.$ in the DNP dataset. For the Octopisa dataset, the largest difference is 480.28m and 10789.69m using XYT and route similarity cohesion, respectively. The route similarity cohesion score varied more compared to the XYT cohesion score. This plot thus suggests the use of smaller values of R.

We now present the effect of the *synchronisation_rate* on the discovered flocks themselves. Figure 1 illustrates how different values of R can affect the flock results by observing the change in the number of moving flocks discovered, the overall flock cohesion, the overall flock separation (based on the spatio-temporal coordinates XYT and route similarity), and the overall silhouette coefficient (XYT and route similarity based).

Figure 1 shows the plots obtained for the *synchronisation_rate* in the two datasets. In general, fewer flocks are found as R increases. Furthermore, the overall flock cohesion varies slightly for different R values, while the overall flock separation tend to decrease with increasing R values. In the case of $R = 8mins.$ in Octopisa and $R = 13mins.$ in DNP, the discovered flocks becomes 0 and hence, the cohesion and separation scores are no longer applicable. Considering the plots for the XYT and route similarity separation scores, it is advisable to set R to a value less than $6mins.$ in DNP and a value less than $4mins.$ in Octopisa due to the sudden drop in the separation scores. A sudden drop occurs when no flock or only a single flock is discovered, or when the distance among the discovered flocks is small. Very large XYT separation scores indicates that the flocks are temporally disjoint. On the other hand, very large route similarity scores indicates that different flocks are following different routes. Finally, the silhouette coefficients summarize the effect of R on both the cohesion and separation scores. The silhouette coefficients are generally close to 1 (i.e., ideal

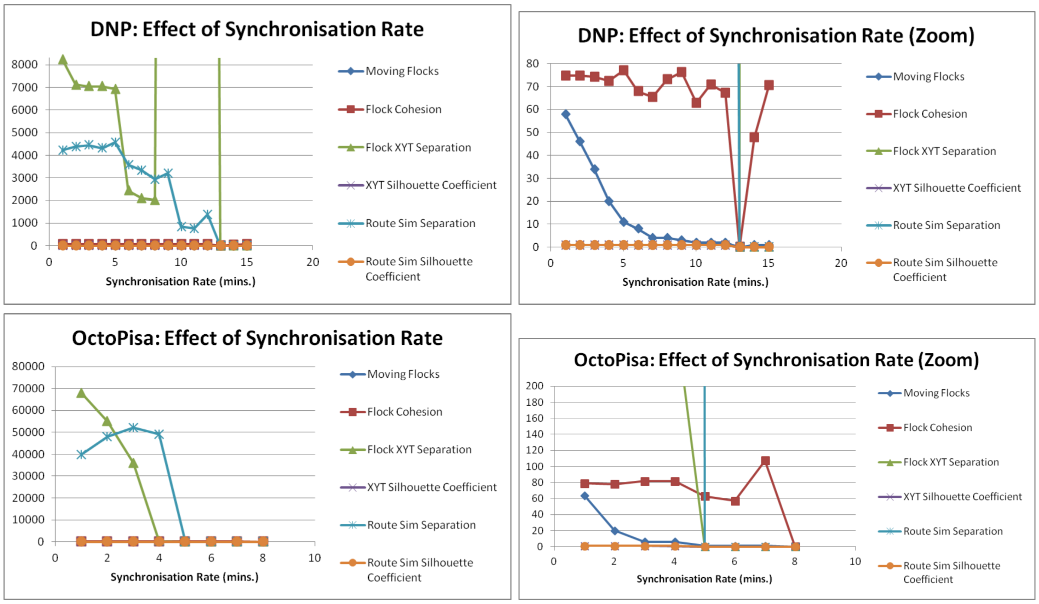


Fig. 1. Effect of the *synchronisation_rate* parameter for the two datasets. We have the full plots on the left part and the zoom in on the right part. The zoom in figures show the variation in the measures that have very small scores compared to the XYT and route similarity separation scores.

case), except for cases wherein the silhouette coefficient is 0. These cases refer to instances wherein only a single flock or no flock was found, making the silhouette coefficient inapplicable.

Effect of the *min_time_slices* (ΔT) Parameter *min_time_slices* and *synchronisation_rate* are parameters that are both related to time. Since the plots and observations for these parameters are generally similar, we no longer present the plots for *min_time_slices*.

As observed with *synchronisation_rate*, an increasing value of ΔT results in fewer number of discovered moving flocks, and, generally, lowering the XYT and route similarity separation scores. The XYT- and route similarity-based silhouette coefficients are either close to 1 when more than a single flock is found, or 0, otherwise. Based on the experiments, a value of 2 or 3 time slices is ideal for the DNP dataset since there is a large drop in the XYT separation score when $\Delta T = 4$. Same is true for the OctoPisa dataset.

Effect of the *min_points* (κ) Parameter Recall that another parameter of the flock algorithm is *min_points*, which refers to the minimum objects that should consist a flock. Based on these experiments, we conclude that the selection of minimum number of points is the most trivial since a large value for min points tends to produce no flock or few flocks; it is a tradeoff between having more flocks but with fewer members, or having few flocks but with more members.

Effect of the *radius* (ϵ) Parameter Lastly, we have also observed the effect of the *radius* parameter, which defines the spatial closeness among flock members. As observed in Figure 2, the number of flocks generally increases as ϵ increases. Meanwhile, the flock cohesion degrades (i.e., intra-distance increases) as ϵ increases. The XYT flock separation score tends to improve as ϵ increases when excluding the cases wherein the discovered flocks do not overlap in time (i.e., maximum XYT separation score is obtained) or no flocks were found. Meanwhile, the route similarity separation score generally improves as larger values of ϵ are used. As with previously observed parameters, the silhouette coefficients for varying ϵ remains close to 1.

The effect of *radius* as compared with the effect of the other parameters is as follows:

1. Out of all the scores used in assessing the effects of the parameters, the number of moving flocks has been the most sensitive. Generally, its value is directly proportional to the value of ϵ while it is inversely proportional to the other parameters. It is also worth noting that a higher number of moving flocks does not necessarily mean that the obtained flock results is better since the quality of the moving flocks may decrease when there are too many flocks discovered.
2. Compared to other flock validity measures, we consider flock cohesion as most important since it is in harmony with and explicit in the definition of a flock (i.e., a flock consists of members that are spatially close together over a specific time duration). While the flock cohesion score linearly increases (i.e., flock cohesion degrades) as the ϵ increases, the cohesion score did not change as much with respect to changing values of the other parameters. Thus, we can conclude that the *radius* has a larger impact on the obtained flock results compared to the other parameters. Excluding the cases wherein no flocks are discovered (i.e., the flock separation score is irrelevant) and the cases wherein there is no overlap in time among the discovered flocks (i.e., the XYT separation score is set to the maximum), higher ϵ generally improves the separation scores whereas higher values for the other parameters generally degrades the separation scores.
3. As a final point, the silhouette coefficient scores obtained by varying different parameters for both datasets were consistently close to 1, except for cases where less than 2 flocks were found.

Based on these experiments, we conclude that (1) The selection of minimum number of points is most trivial since a large value for *min_points* tends to a few flocks, if any at all; it is a tradeoff between having more flocks but with fewer members, or having few flocks but with more members. We also conclude that (2) the most crucial parameter is the *radius*, since it exhibited a larger effect on the flock cohesion score compared to the other parameters. Lastly, we conclude that (3) while *radius* is the most crucial parameter, it is still important to choose good values for the other parameters since they still affect the quality of the discovered flocks.

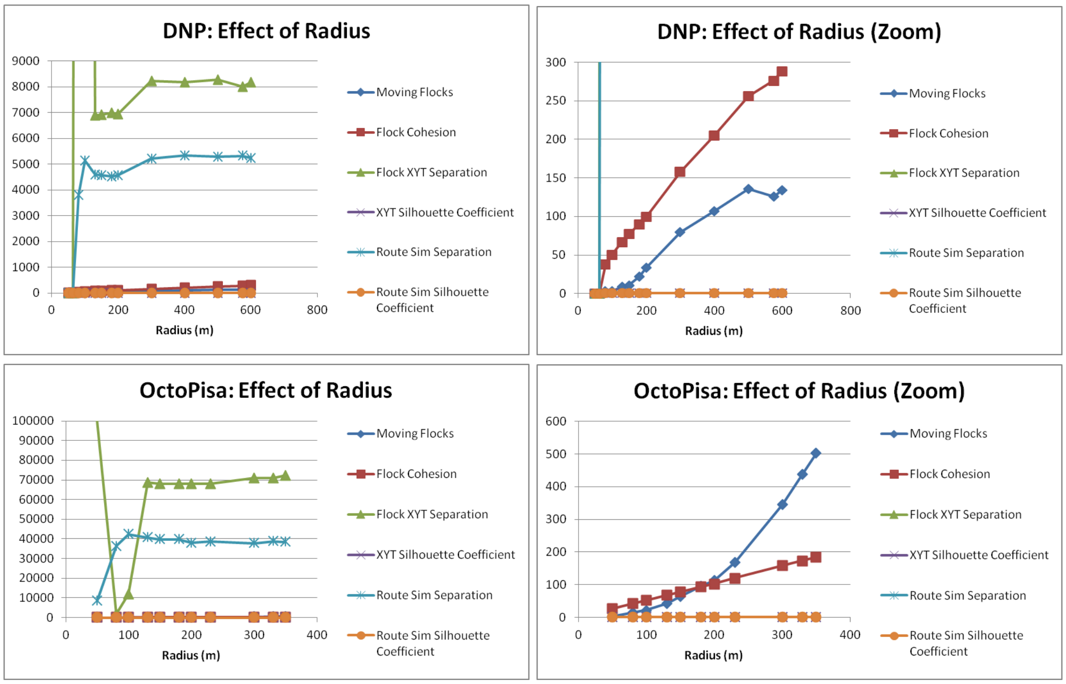
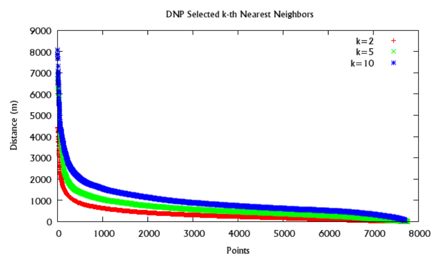


Fig. 2. Effect of the *radius* parameter with respect to flock quality measures in the two datasets.

4.3 Finding a Suitable *radius* Value

Since *radius* is a crucial parameter of the flock algorithm, we propose the following technique, which is an extension of the technique introduced for the *Eps* parameter of DBSCAN. Since DBSCAN deals with single n-dimensional data points while the flock algorithm deals with 3D data points (spatial component plus time) that are connected through object IDs, adjustments to their technique are necessary to accommodate the points linked by the same object IDs. The general idea of the extended technique is to compute the k-th distance among objects that co-occur in the same time instant where k is $min\_points - 1$ and k-th distance refers to the distance of a point from its k-th nearest neighbour. Once the k-th distances have been computed for each point, they are sorted in non-ascending order and plotted as a line graph. The portion in which there is a sudden decrease in the k-th distance suggests an upper bound for the *radius* parameter of the algorithm. It is important to note that the trend of the plots in Figure 3 from right to left is as follows: increasing distance leads to the inclusion of more entities as members of discovered flocks. This becomes less apparent once we reach the portion of sudden decrease up to the leftmost part of the plots. Within this range, larger distance may lead to an increase in more entities included members in the discovered flocks, but this increase is very small. In fact, the leftmost part of the plots represent the cases wherein very large distance values no longer results in any increase of the entities. This happens when all entities are already member of one and the same flock since the chosen radius, which is based on the plots' k-th distance, is too large.



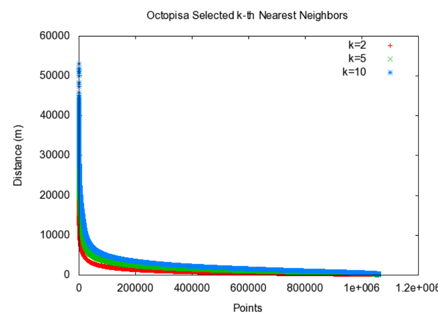


Fig. 3. Plot for k-th nearest neighbours for selected k's in the DNP (left) and the OctoPisa (right) datasets.

Using the top part of Figure 3 for the DNP dataset, a suggested *radius* value should be below the 500m-2000m range for flocks with at least 3 members (i.e., $k = 2$).

The obtained plot for the OctoPisa dataset is shown on the bottom part of Figure 3. It suggests 3000m-4000m as an upper bound for the *radius*. This is reasonable since the OctoPisa dataset covers a wider spatial area (about 4600 Km^2 vs about 48Km^2 of DNP). It is also worth noting that the plots suggest different *radius* values for varying k's and yet, the division between the objects that would be included in some flock and those that are considered as noise is almost the same. Combining the suggested upper bound with contextual knowledge and the observation on the effect of *radius*, we recommend that a good range of values for *radius* is between 80m to 300m for DNP and between 50m to 300m for OctoPisa. Table 1 summarizes the main recommendations for a good range of parameter values for the two datasets.

5 Conclusions and Future Work

This paper provides an empirical evaluation of the effects of parameters in two different moving objects datasets, aimed at delineating a data-driven parameter estimation method for flock mining. We have evaluated the parameter setting

Parameter Name	*OctoPisa*	*DNP*	*Remarks*
κ	2-3	2-3	Prefer higher values but should consider number of discovered flocks, cohesion and separation scores
ΔT	3-4	2-3	Prefer higher values but consider number of discovered flocks, cohesion and separation scores
R	$<$ 4 mins. best: 1-2	$<$ 6 min. best: 1 & 4	Based on XYT separation score
ϵ	50m to 300m	80m to 300m	the DBSCAN-based plot (gives the optimal result in terms of cluster assignment) and the plots on moving flocks, cohesion and separation scores

Table 1. A table summarizing the main suggestions for flock parameters

methods in trajectories of pedestrian moving in a park and GPS data sets of moving vehicles.

We are also studying algorithm validation methods, which has been omitted here for lack of space. We are extending the experiments including additional trajectory datasets to further validate our results and to propose a formal framework for general flock mining parameters evaluation. Future work includes the extension of this approach to mining other kinds of moving objects such as animals.

References

1. Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.
2. Fosca Giannotti and Dino Pedreschi, editors. *Mobility, Data Mining and Privacy - Geographic Knowledge Discovery*. Springer, 2008.
3. M. C. Gonzlez, C. A. Hidalgo, and A.-L. Barabsi. Understanding individual human mobility patterns. *Nature*, 453, 2008.
4. Joachim Gudmundsson and Marc J. van Kreveld. Computing longest duration flocks in trajectory data. In *GIS*, 2006.
5. Hoyoung Jeung, Man Lung Yiu, Xiaofang Zhou, Christian S. Jensen, and Heng Tao Shen. Discovery of convoys in trajectory databases. *Proc. VLDB Endow.*, 1:1068–1080, 2008.
6. Patrick Laube, Stephan Imfeld, and Robert Weibel. Discovering relative motion patterns in groups of moving point objects. *International Journal of Geographical Information Science*, 19(6):639–668, 2005.
7. Monica Wachowicz, Rebecca Ong, Chiara Renso, and Mirco Nanni. Finding moving flock patterns among pedestrians through collective coherence. *International Journal of Geographical Information Science*, 25(11):1849–1864, 2011.