Trajectory Distances

Di Yao

Outline

- Point based Distance
  - Euclidean, DTW, LCSS, EDR
- Distance based Distance
  - Hausdorff distance, Frechet distance
- Shape based Distance
  - Procrustes Distance, Canonical Warping Distance
- Segment based Distance
  - One Way Distance, LIP distance
- Task Specific Distance
  - TRACLUS, road, semantic, grid
- Conclusion

Euclidean Distance

- Let L_i and L_j are p-dimensional trajectory segments with length of n

$$D_E(L_i, L_j) = \frac{1}{n} \sum_{k=1}^{n} \sqrt{\sum_{m=1}^{p} (a_k^m - b_k^m)^2}$$

- Pons
  - Linear computing time
- Cons
  - Trajectories must be the same length

Dynamic Time Warping(DTW)

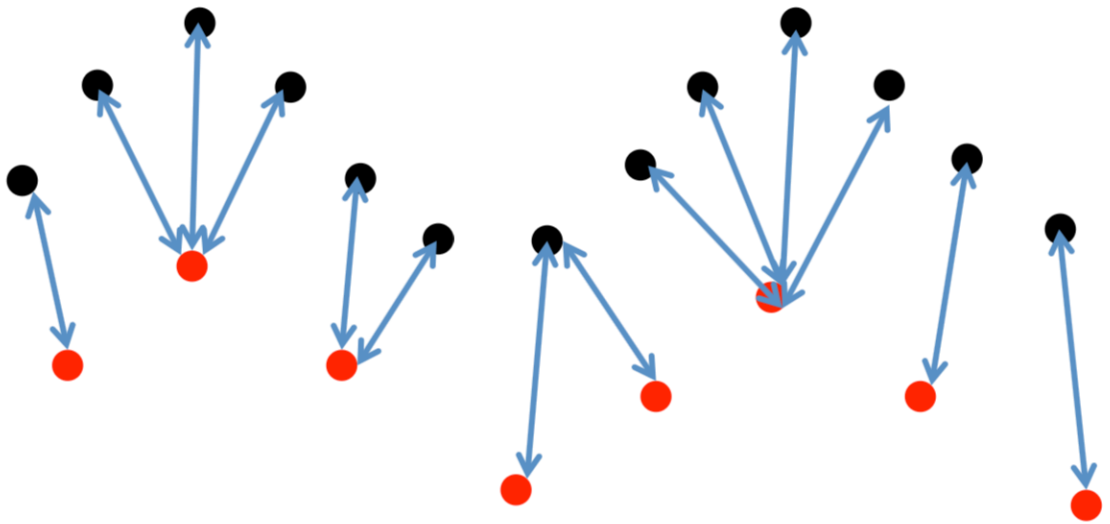
- Adaptation from time series distance measure
- Goal: Its goal is to find the warping path w between two trajectories with the smallest warping cost
  - minimize the aggregate distance between matched points

$$D_D(L_i, L_j) = \begin{cases} 0 & m = n = 0 \\ \infty & m = 0 || n = 0 \\ dist(a_i^k, b_j^k) + \min \begin{cases} D_D(Rest(L_i), Rest(L_j)) \\ D_D(Rest(L_i), L_j) \\ D_d(L_i, Rest(L_j)) \end{cases} & others \end{cases}$$

Keogh, E. J., "Exact Indexing of Dynamic Time Warping," In Proc. 28th Int'l Conf. on Very Large Data Bases, Hong Kong, China, pp. 406–417, Aug. 2002.

Dynamic Time Warping(DTW)

- Pros
  - Time differences are taken into account
  - Provides a better match than Euclidean measure
- Cons
  - DTW distance is sensitive to noise



Longest Common Sub-Sequence(LCSS)

- Adaptation of string similarity
  - LCSS('abcde','bd') = 2
- Threshold-based equality relationship
  - Two locations are regarded as equal if they're 'close' (compared to a threshold)

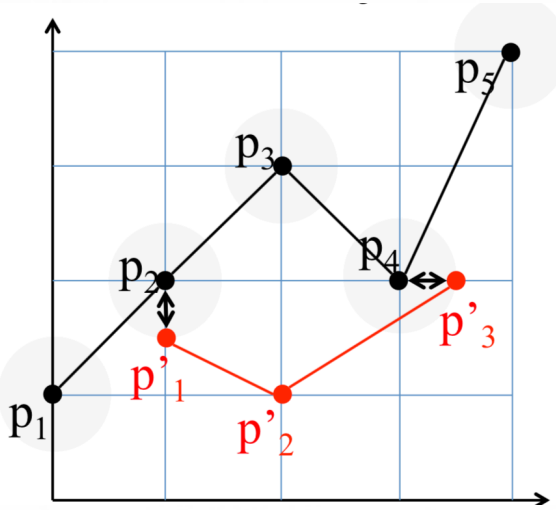
$$D_L(L_i, L_j) = \begin{cases} 0 & n = m = 0 \\ 1 + LCSS_{\sigma,\varepsilon}(Head(L_i), Head(L_j)) & |a_i^x - b_j^x| \le \sigma, and \left|a_i^y - b_j^y\right| \le \varepsilon \\ \max(LCSS_{\sigma,\varepsilon}(Head(L_i), L_j), \\ \quad LCSS_{\sigma,\varepsilon}(L_i, Head(L_j))) & others \end{cases}$$

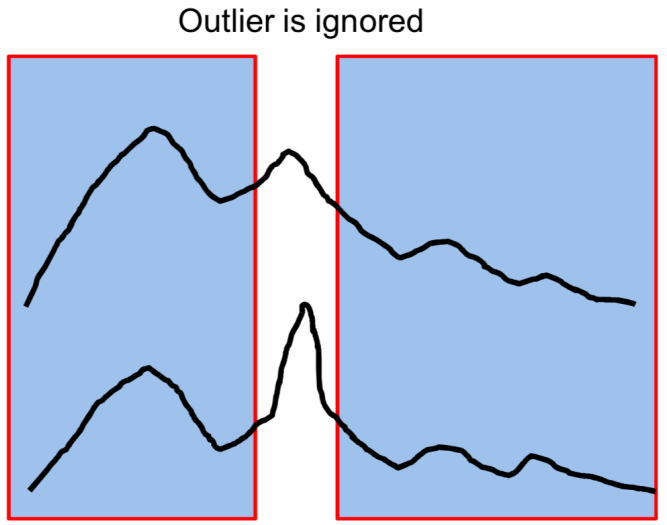
Michail V, Marios H, Dimitrios G (2006) Indexing multidimensional time-series. VLDBJ 15(1):1–20

Vlachos M, Kollios G, Gunopulos D. Discovering similar multidimensional trajectories. *Proc 18th Int Conf Data Eng*. 2002:673-684. doi:10.1109/ICDE.2002.994784.

Longest Common Sub-Sequence(LCSS)

- Pros
  - Not sensitive to noise
- Cons
  - Hard to define threshold





Edit Distance on Real Sequence (EDR)

- Adaptation from Edit Distance on strings
  - Number of insert, delete, replace needed to convert A into B
- Threshold-based equality relationship(Like LCSS)

Definition 3: Using a matching threshold $\varepsilon \geq 0$, and trajectories T_A and T_B with lengths m and k, the EDR_ε value (edit distance) defined in Chen et al. [6] is adapted to the following:

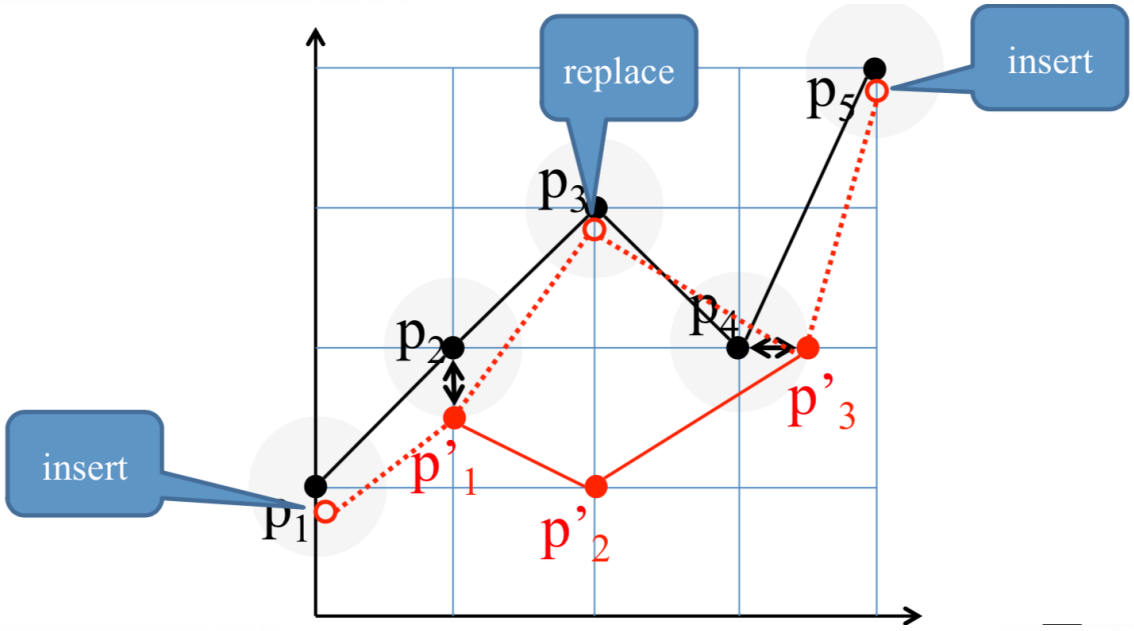
$$EDR_\varepsilon(T_A, T_B) = \begin{cases} k, & \text{if } m = 0 \\ m, & \text{if } k = 0 \\ min(EDR_\varepsilon(Rest(T_A), Rest(T_B)) + subcost, & \\ \quad EDR_\varepsilon(Rest(T_A), T_B) + 1, & otherwise \\ \quad EDR_\varepsilon(T_A, Rest(T_B)) + 1), & \end{cases}$$

subcost = 0 if the first point of TA lies within the matching threshold of the first pointof TB in every dimension, and subcost = 1 otherwise.

ChenL, ÖzsuM,OriaV(2005) Robust and fast similarity search formoving object trajectories. In: Proceedings of the 2005ACM SIGMOD international conference on management of data,ACM,NewYork,NY,USA, pp 491–502

Edit Distance on Real Sequence (EDR)

- Pros
  - Insensitive to noise
- Cons
  - Hard to define threshold



Relation Between EDR and LCSS

- They are both count-based
  - LCSS counts the number of matched pairs
  - EDR counts the cost of operations needed to fix the unmatched pairs
- Higher LCSS, lower EDR
  - If cost(replace) = cost(insert) = cost(delete)
  - EDR(X,Y) = L(X)+L(Y) – 2LCSS(X,Y)

Outline

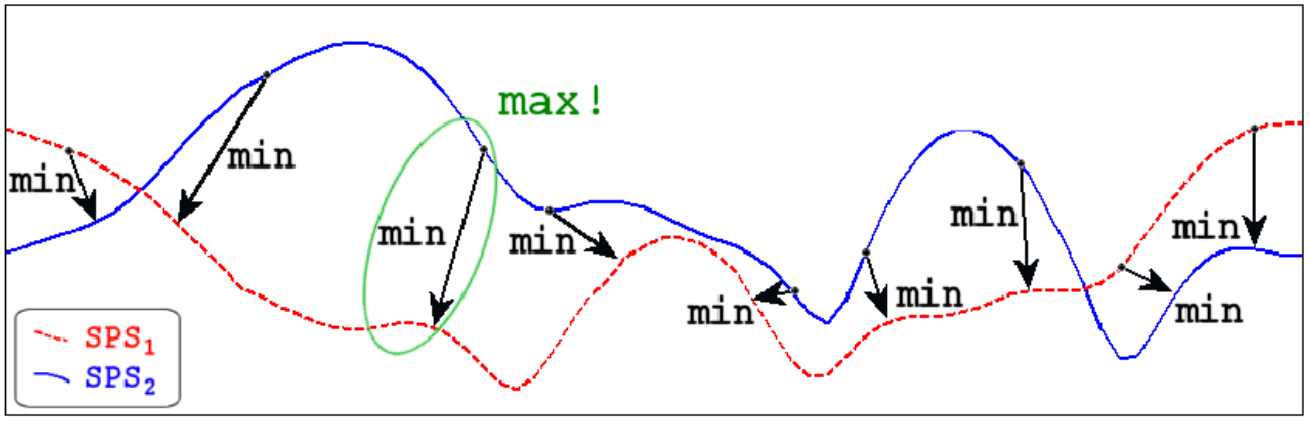
- Point based Distance
  - Euclidean, DTW, LCSS, EDR
- Shape based Distance
  - Hausdorff distance, Frechet distance
- Segment based Distance
  - One Way Distance, LIP distance
- Task Specific Distance
  - TRACLUS, road, semantic, grid
- Conclusion

Hausdorff distance

- Used to measure how far two trajectories are from each other

$$D_H\left(L_i, L_j\right) = \max(h(L_i, L_j), h(L_j, L_i))$$
$$\text{where } h(L_i, L_j) = \max_{a \in L_i}(\min_{b \in L_j}(dist(a, b)))$$

- Pros
  - M… …veen tw…
- Cons
  - Sensitive to noise data

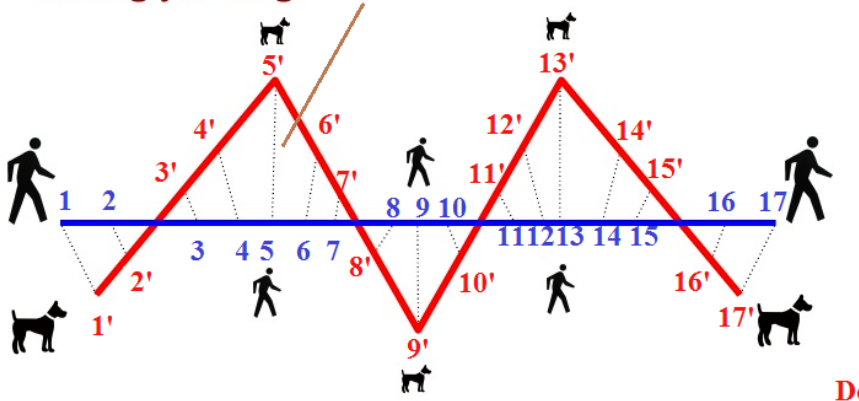


H. A It, "The computational geometry of comparing shapes, " in EffiCient Algorithms. Springer, 2009, pp. 235-248

Xie D, Li F, Phillips JM. Distributed Trajectory Similarity Search. *Pvldb*. 2017:1478-1489.

Frechet Distance

- Sh
- Dc
  -
  -
  -



Note: the number of points of two curves are not necessary the same.
Now I want to save all points of each curve in the order they appear in each shape.
Assuming these number in the picture represent points along the shape with coordinate x, y.
Now I want to save them into two array with this order as follows:

Dog array	1'	2'	3	...	17'

Owner array	1	2	3	...	17

How can I do that? Assuming that the points are sampling from two curves. We already know the coordinate of each point but they are unordered.

$$\text{where, } \|C\| = \max_{k=1}^{K} \text{dist}(a_i^k, b_j^k)$$

$D_F(L_i, L_j)$ is the Fréchet distance between trajectory segments L_i and L_j. Here, L_i and L_j are the trajectory segments whose lengths are m and n respectively. K $= min(m, n)$. a_i^k and b_j^k are the kth points on trajectory segments L_i and L_j respectively. $dist(a_i^k, b_j^k)$ is the Euclidean distance between a_i^k and b_j^k.

H. A It, "The computational geometry of comparing shapes, " in EffiCient Algorithms. Springer, 2009, pp. 235-248

Xie D, Li F, Phillips JM. Distributed Trajectory Similarity Search. *Pvldb*. 2017:1478-1489.

Comparison

Measurement	Parameters	Applicable scope	Anti-noise property	Computational complexity
Euclidean distance	Parameter-free	The length of two trajectories must be the same	Weakest	$O(n)$
PCA + Euclidean distance	Parameter-free	The length of two trajectories must be the same	Weaker	$O(n)$
Hausdorff distance	Parameter-free	It is applicable for most of trajectory data	Weaker	$O(m*n)$
LCSS distance	σ and ε (distance threshold of x and y direction)	It is applicable for most of trajectory data except the discrete trajectory data	Strong	$O(m*n)$
DTW distance	Parameter-free	Trajectory must be continuous and there does not exist completely dissimilar trajectory range in trajectories	Weaker	$O(m*n)$
Fréchet distance	Parameter-free	Trajectory data is discrete or continuous	Weaker	$O(m*n)$ $O(mn \cdot \log(mn))$

Can be optimized to O(m+n) in a convx setting

Comparison

- Evaluation Method
  - Given a set of labeled trajectories T
  - Utilize one nearest neighbor (1NN) classifier to evaluate
    - Underlying distance metric is critical to the performance of 1NN classifier
    - 1NN classifier is parameter free

Ding H, Trajcevski G, Scheuermann P, Wang X, Keogh E. Querying and Mining of Time Series Data : Experimental Comparison of Representations and Distance Measures. 2008.

Comparison

- Evaluation Method

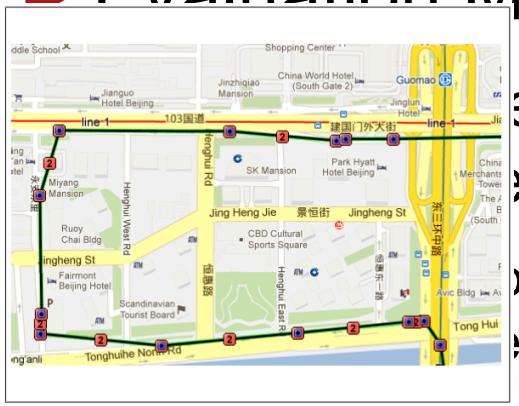


Figure 2: Increase Sampling Rate Transformation Function

Figure 3: Decrease Sampling Rate Transformation Function

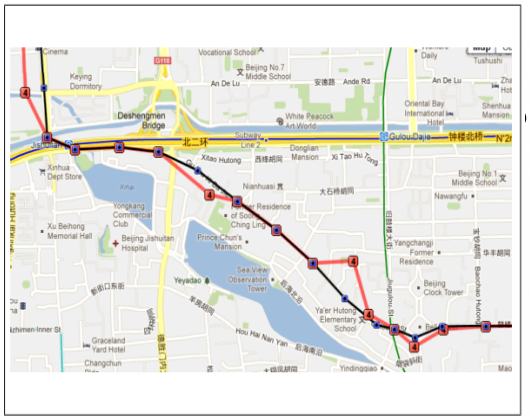


Figure 4: Random Shift Transformation Function

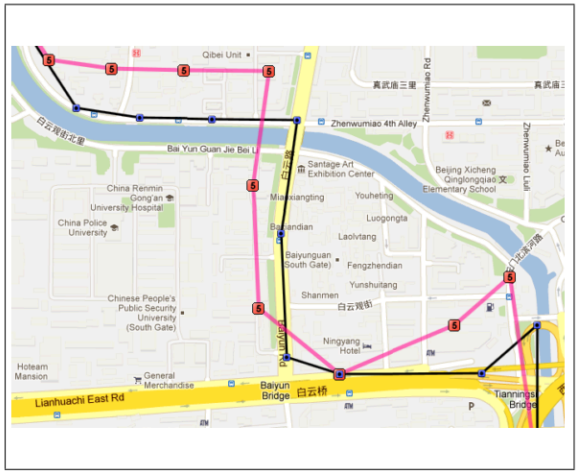


Figure 5: Synchronized Shift Transformation Function

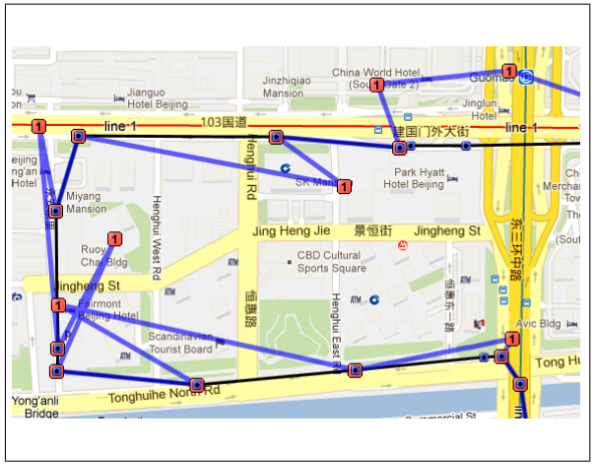


Figure 6: Add Noise Transformation Function

Wang H, Sadiq S. An Effectiveness Study on Trajectory Similarity Measures(ADC 2013).pdf. 2013;137(February):13-22.

Outline

- Accumulation based Distance
  - Euclidean, DTW, LCSS, EDR
- Point based Distance
  - Hausdorff distance, Frechet distance
- Segment based Distance
  - One Way Distance, LIP distance
- Task Specific Distance
  - TRACLUS, road, semantic, grid
- Conclusion

One Way Distance

- One Way Distance of trajectory T1 and T2:
  - Integral of the distance from points of T1 to T2, divided by the length of T1

$$D_{\text{owd}}(T_1, T_2) = \frac{1}{|T_1|}\left(\int_{p \in T_1} D_{\text{point}}(p, T_2)\, dp\right)$$

$$D_{\text{point}}(p, T) = \min_{q \in T} D_{\text{Euclid}}(p, q)$$

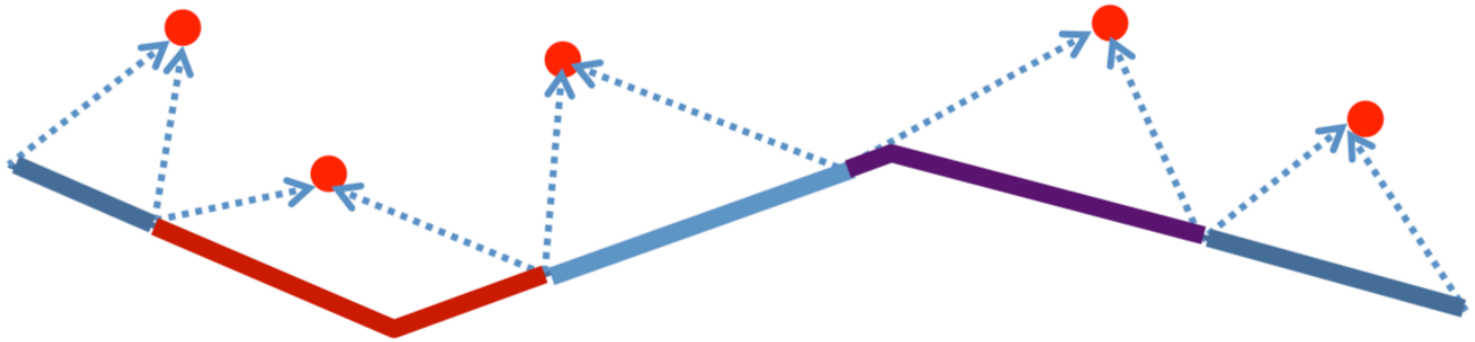
  - Symmetric measure

$$D(T_1, T_2) = \frac{1}{2}\left(D_{\text{owd}}(T_1, T_2) + D_{\text{owd}}(T_2, T_1)\right)$$

Bin Lin, Jianwen Su, One Way Distance: For Shape Based Similarity Search of Moving Object Trajectories. In Geoinformatica (2008)

One Way Distance

- Consider one trajectory as piece-wise line segment, and the other as discrete samples

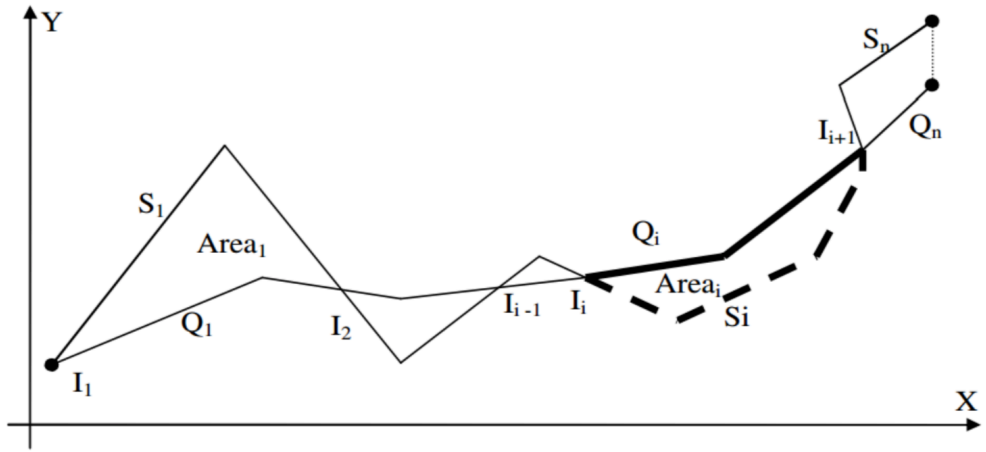


LIP distance

- Locality In-between Polylines

$$LIP(Q,S) = \sum_{\forall\ polygon_i} Area_i \cdot w_i \qquad w_i = \frac{Length_Q(I_i, I_{i+1}) + Length_S(I_i, I_{i+1})}{Length_Q + Length_S}$$

  - *Polygon* is the set of polygons formed between intersection points(Only work for 2-dimensional trajectories)



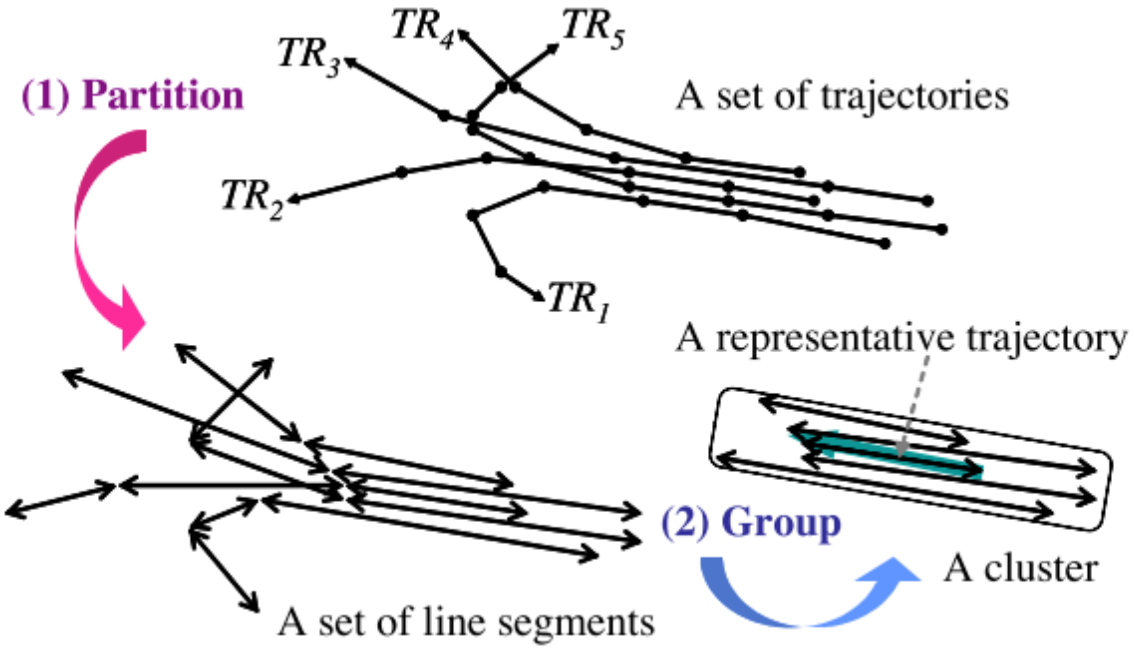
Nikos Pelekis et al, Similarity Search in Trajectory Databases. Symposium on Temporal Representation and Reasoning 2007

Outline

- Accumulation based Distance
  - Euclidean, DTW, LCSS, EDR
- Point based Distance
  - Hausdorff distance, Frechet distance
- Segment based Distance
  - One Way Distance, LIP distance
- Task Specific Distance
  - TRACLUS, road, clue, semantic, grid
- Conclusion

TRACLUS

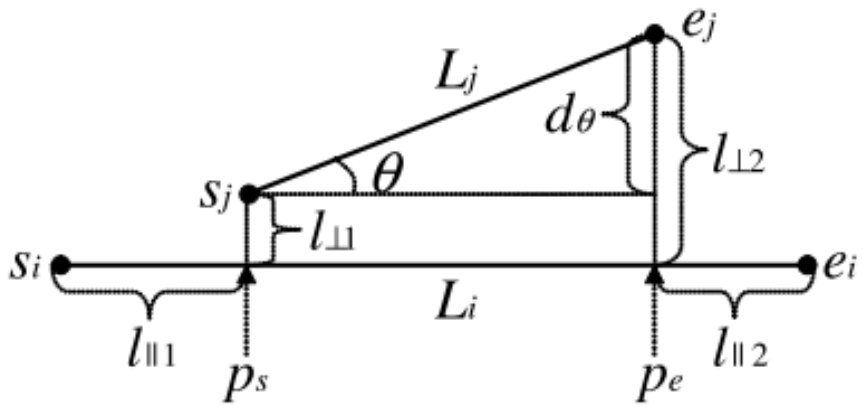
- A trajectory clustering approach that considers *sub-trajectories*
  - Parts of trajectories might match even when the trajectory as a whole does not



Lee J-G, Han J, Whang K-Y. Trajectory clustering. *Proc 2007 ACM SIGMOD Int Conf Manag data - SIGMOD '07*. 2007:593. doi:10.1145/1247480.1247546.

TRACLUS

- Define perpendicular, parallel and angle distances for segment lines
- Then, weight sum the distances as the distance of segment



$$d_\perp = \frac{l_{\perp 1}^2 + l_{\perp 2}^2}{l_{\perp 1} + l_{\perp 2}}$$

$$d_\parallel = \mathrm{MIN}(l_{\parallel 1}, l_{\parallel 2})$$

$$d_\theta = \|L_j\| \times \sin(\theta)$$

TRACLUS

- Partition: Trajectory to Sub-trajectory
  - minimum description length (MDL)

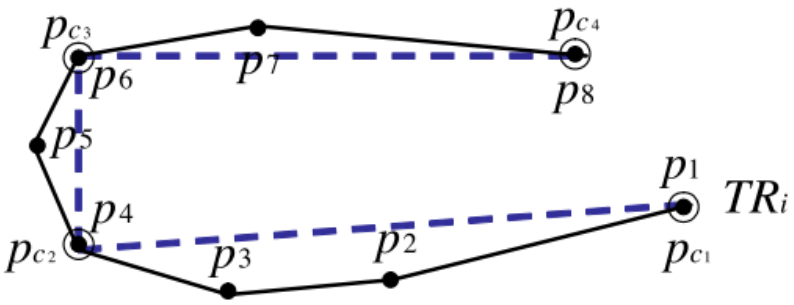
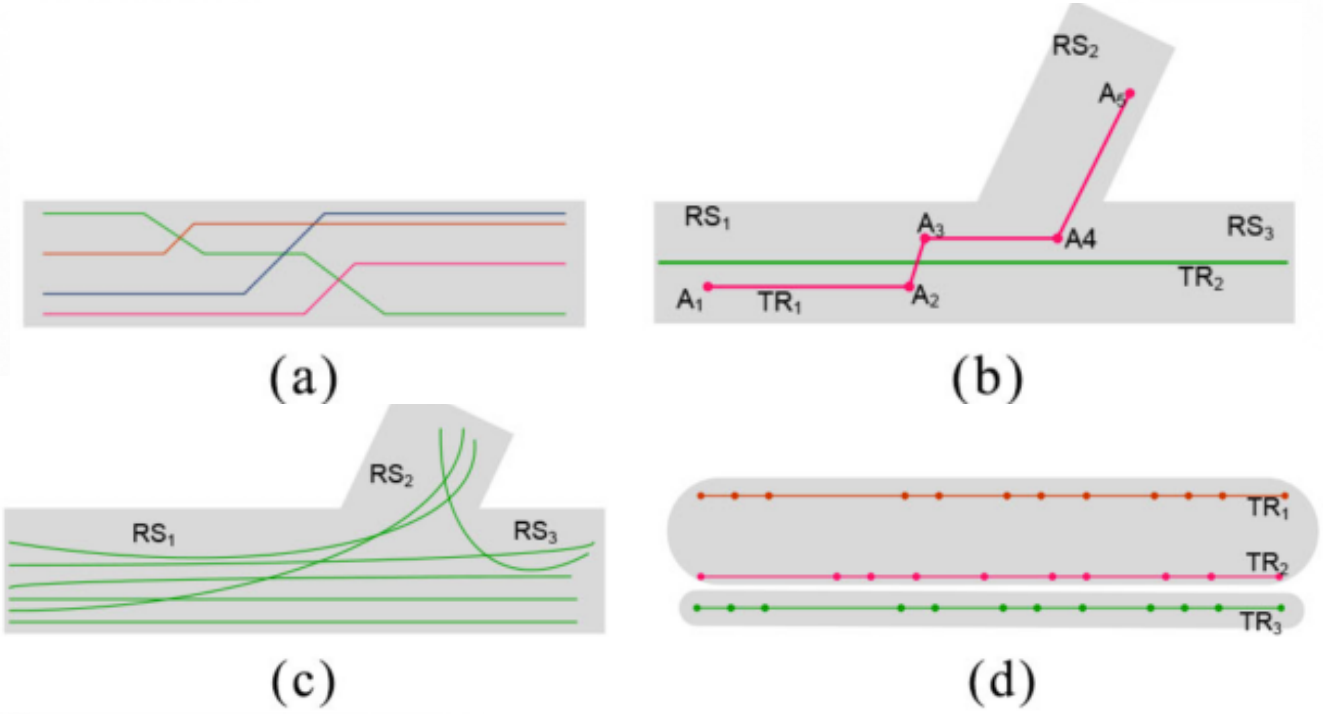


Figure 6: An example of a trajectory and its trajectory partitions.

NEAT-road network aware

- Pervious works on trajectory do not consider the road network factor which leads to bad trajectory clustering



Han B, Liu L, Omiecinski E. Road-network aware trajectory clustering: Integrating locality, flow, and density. *IEEE Trans Mob Comput*. 2015;14(2):416-429. doi:10.1109/TMC.2013.119.

NEAT-road network aware

- Three Phase trajectory clustering
  - Base Cluster Formation
  - Flow cluster formation
  - Flow Cluster Refinement

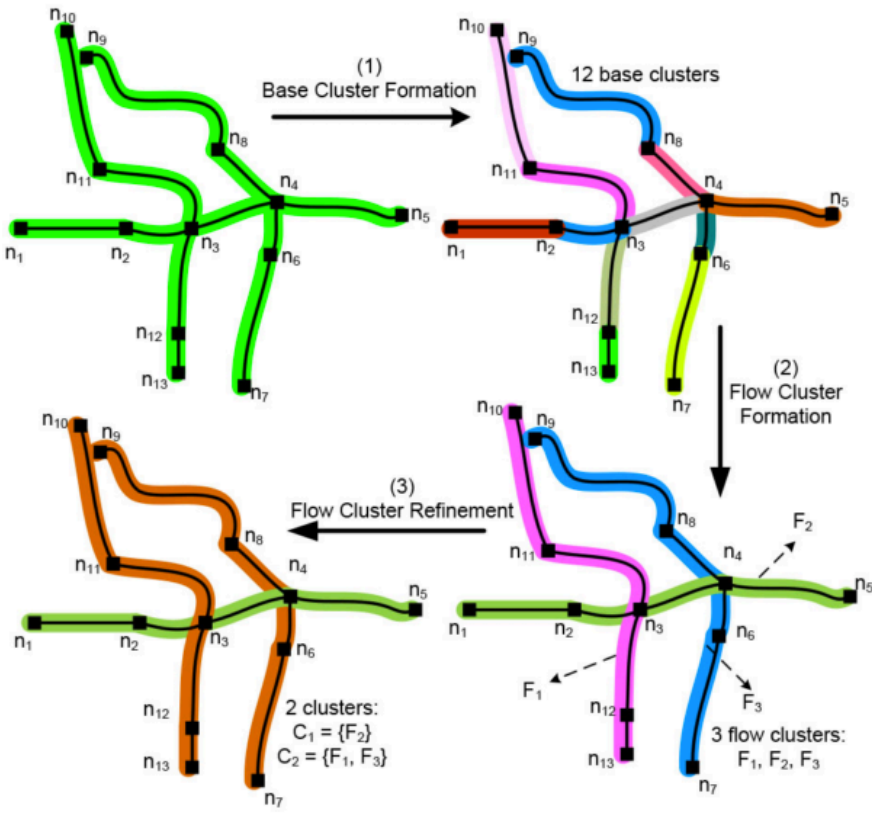


Fig. 3. Example of three phase clustering in the NEAT framework.

NEAT-road network aware

- Three Phase trajectory clustering
  - Base Cluster Formation
    - Segment trajectory by road network
    - Assign each fragment to a road segment by map matching
  - Flow cluster formation
    - Choose the density core as initial cluster S
    - Merge the neighbor cluster into S by computing flow factor q, density factor k and speed limit factor v
    - Weight sum above factors and using a threshold to merge cluster
  - Flow Cluster Refinement
    - Calculate the shortest path base Hausdorff distance between clusters and then Optimization clusters by DBSCAN

NEAT-road network aware

Definition 5 (netflow). *The netflow between two base clusters S_i and S_j, denoted by $f(S_i, S_j)$, is the number of trajectories participating in both clusters: $f(S_i, S_j) = |PTr(S_i) \cap PTr(S_j)|$ The function netflow between two base clusters computes the number of common objects traveled on both representative road segments e^{S_i} and e^{S_j}.*

Definition 6 (f-neighborhood). *Let B denote a set of base clusters, S_i denote a base cluster and n_u denote one endpoint of e^{S_i}. The f-neighborhood of S_i wrt. n_u, denoted by $N_f(S_i, n_u)$, is the set of base clusters that have at least one common participating trajectory, and is formally defined as: $N_f(S_i, n_u) = \{S_j |\ e^{S_j} \in L_{n_u}(e^{S_i})\ \&\ f(S_i, S_j) > 0\}$.*
Let n_v be the other endpoint of e^{S_i}. We define the f-neighborhood of S_i wrt. e^{S_i} as: $N_f(S_i) = N_f(S_i, n_u) \cup N_f(S_i, n_v)$. Each $S_j \in N_f(S_i)$ is called the f-neighbor of S_i. Note that the f-neighbor is a symmetric relation.

Definition 9. *Given a base cluster S and n_u as one endpoint of road segment e^S, the flow factor q, density factor k and speed limit factor v of a base cluster $S_j \in N_f(S, n_u)$ wrt. S are defined respectively as follows:*

$$q = f\left(S, S_j\right) / |PTr(S)| \tag{1}$$

$$k = d\left(S_j\right) / (d(S) + \sum_{S_i \in N_f(S, n_u)} d\left(S_i\right)) \tag{2}$$

$$v = speed\left(S_j\right) / \sum_{S_i \in N_f(S, n_u)} speed\left(S_i\right) \tag{3}$$

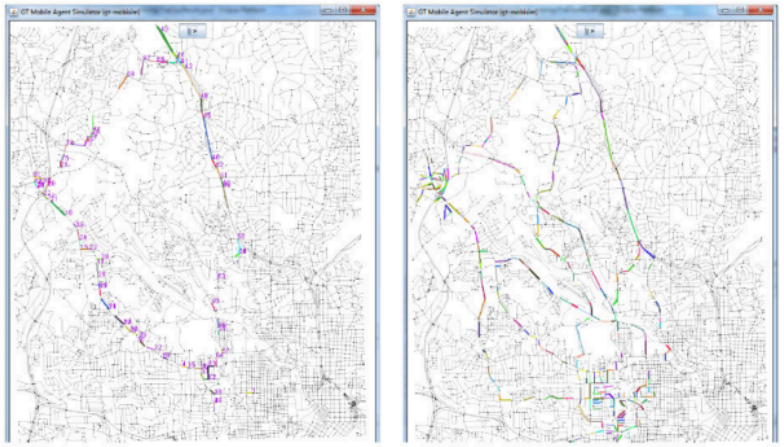
where $speed(S_j)$ is the speed limit of e^{S_j}.

Definition 10. *Given a base cluster S and n_u as one endpoint of e^S, the merging selectivity of a base cluster $S_j \in N_f(S, n_u)$ is defined as:*

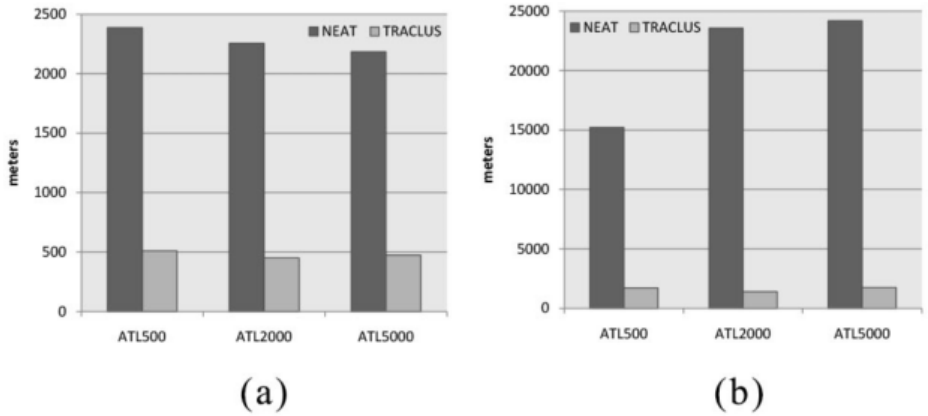
$$SF(S, S_j) = w_q \times q + w_k \times k + w_v \times v \tag{4}$$

where the coefficients w_q, w_k and w_v determine the weights of q, k, v respectively. The weights $w_q \geq 0$, $w_k \geq 0$ and $w_v \geq 0$ satisfy $w_q + w_k + w_v = 1$.

NEAT-road network aware

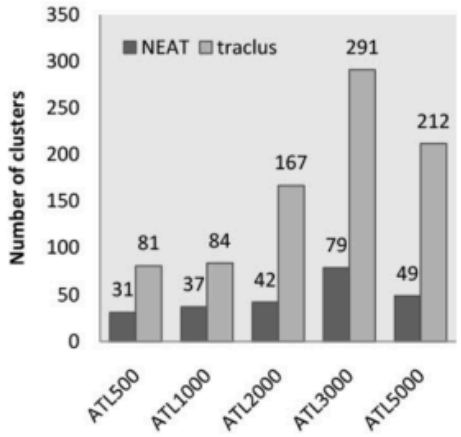


(a) (b)

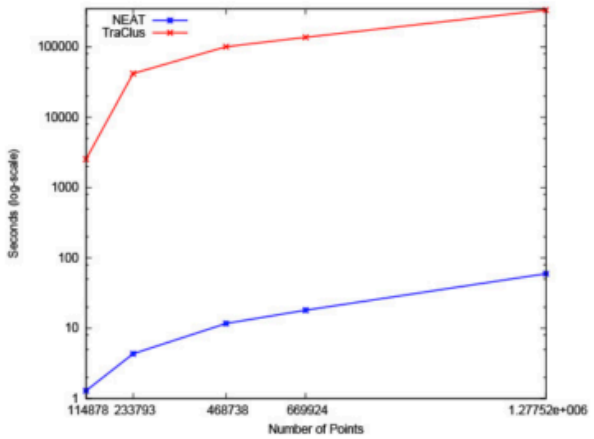


(a) (b)

Fig. 7. TraClus. (a) 81 clusters for ATL500 (ε=10m, *MinLns*=30). (b) 460 clusters for ATL500 (ε=1m, *MinLns*=1).



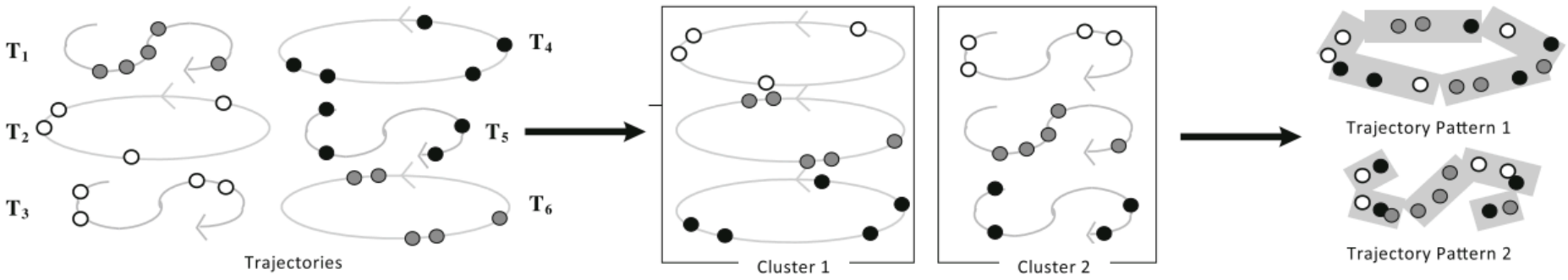
(c)



(d)

Clue-aware trajectory similarity

- "clues" referring to those spatially and temporally co-located data points among trajectories
- The concept of clues is to evaluate how many such co-located points exist between two trajectories
  - closer points → stronger clues
  - co-located points reveal clues → occurrence times close(tolerate such temporal shifting)



Hung CC, Peng WC, Lee WC. Clustering and aggregating clues of trajectories for mining trajectory patterns and routes. *VLDB J*. 2015;24(2):169-192. doi:10.1007/s00778-011-0262-6.

Clue-aware trajectory similarity

- Clue-aware trajectory similarity

Definition 4 Spatial Decaying Function: Given a spatial threshold ϵ and two data points $p_{i,\ell} = (l_{i,\ell}, t_{i,\ell})$ and $p_{j,k} = (l_{j,k}, t_{j,k})$ from two trajectories (i.e., T_i and T_j), a spatial decaying function for two points $p_{i,\ell}$ and $p_{j,k}$ is defined as

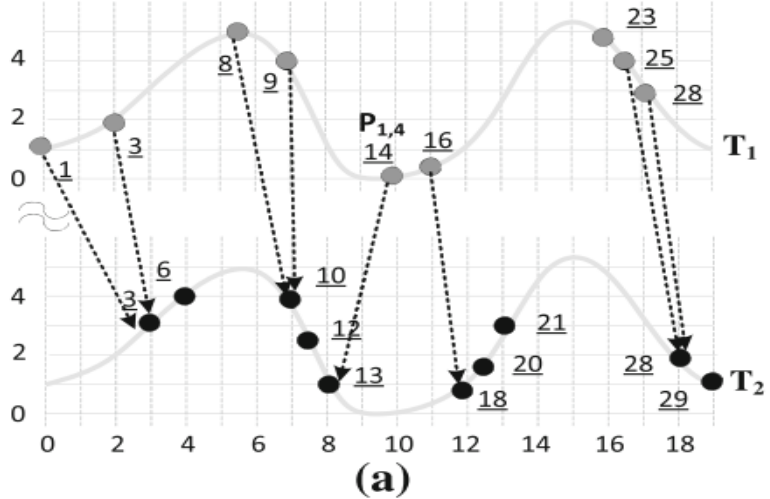
$$f_\epsilon(p_{i,\ell}, p_{j,k}) = \begin{cases} 0 & , \text{if } \text{dist}(p_{i,\ell}, p_{j,k}) > \epsilon \\ 1 - \frac{\text{dist}(p_{i,\ell}, p_{j,k})}{\epsilon} & , \text{otherwise} \end{cases}$$

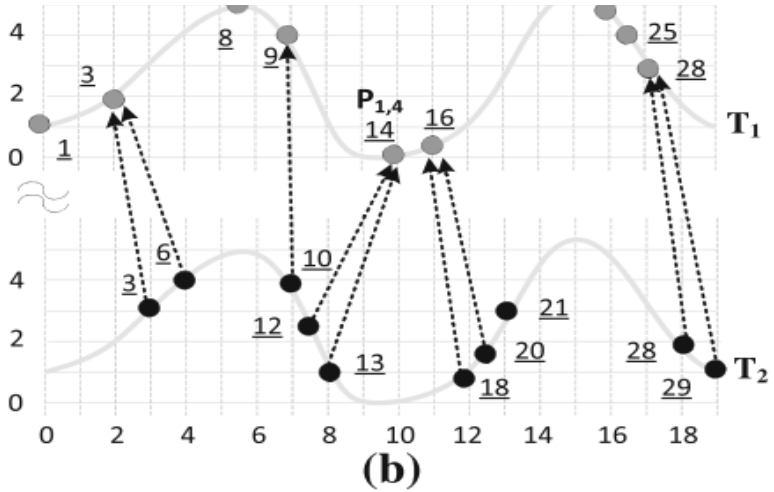
where $\text{dist}(\cdot, \cdot)$ denotes Euclidean distance between two data points.

Definition 5 Clue score of data points: Given a point $p_{i,\ell}$, a reference trajectory T_j, a spatial threshold ϵ, and a temporal threshold τ, the clue score of data point $p_{i,\ell}$ to trajectory T_j is defined as $score_{\epsilon,\tau}(p_{i,\ell}, T_j) = \max\{f_\epsilon(p_{i,\ell}, p_{j,k}) | p_{j,k} \in T_j \text{ and } t_{j,k} \in [t_{i,\ell} - \tau, t_{i,\ell} + \tau]\}$.

Definition 6 Clue-aware trajectory similarity: Given a spatial threshold ϵ and a temporal threshold τ, the clue-aware trajectory similarity from T_i to T_j is defined below:

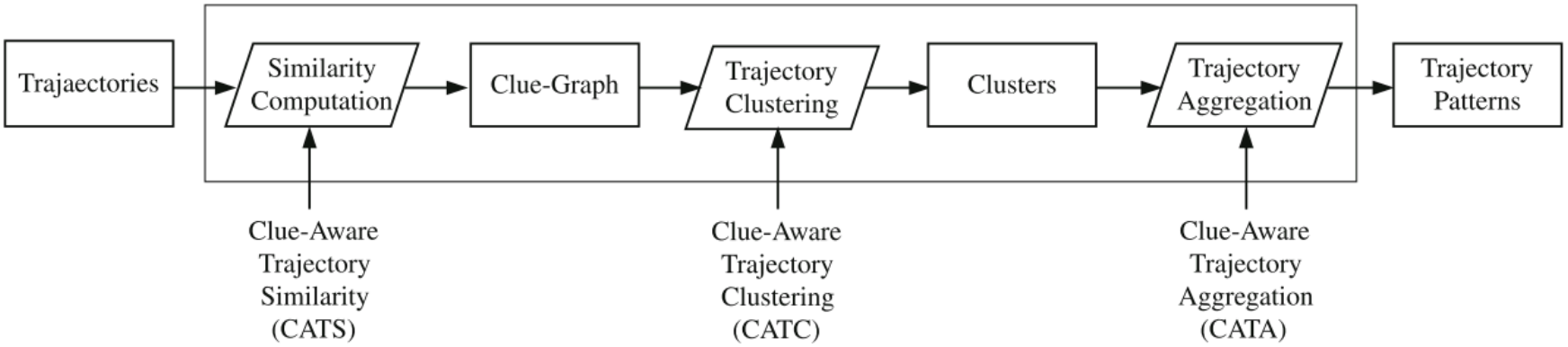
$$CATS_{\epsilon,\tau}(T_i, T_j) = \frac{1}{|T_i|} \times \sum_{p_{i,\ell} \in T_i} score_{\epsilon,\tau}(p_{i,\ell}, T_j).$$

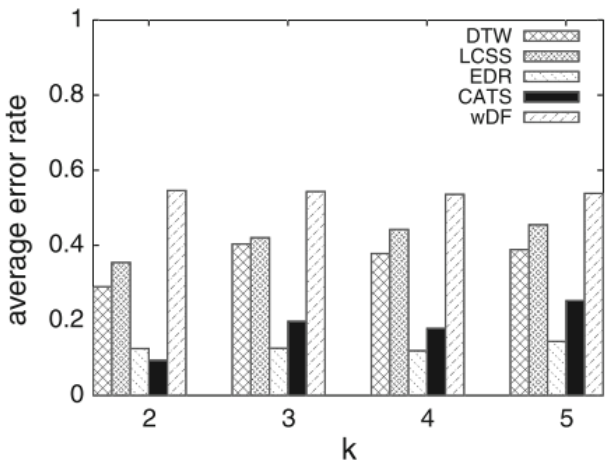




Clue-aware trajectory similarity

- Clue-aware trajectory clustering





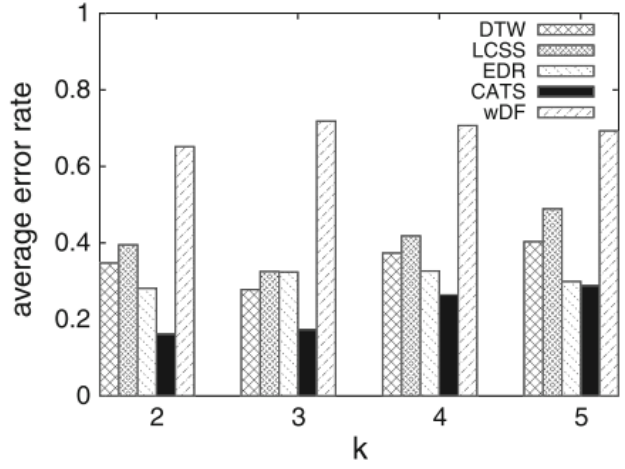


Fig. 12 Average error rates with k varied. **a** 3 types. **b** All types

Semantic Trajectory

- Semantic Trajectory: trajectory sequence includes spatial, temporal, semantic information
  - Semantic information→ a set of key words
- Similarity Computation
  - Combine the text similarity and geographical distance together
- Experiment
  - Time cost per round (seconds)

Zheng B, Yuan NJ, Zheng K, Xie X, Sadiq S, Zhou X. Approximate keyword search in semantic trajectory database. In: *Proceedings - International Conference on Data Engineering*. Vol 2015-May. ; 2015:975-986. doi:10.1109/ICDE.2015.7113349.

Q&A !

Di Yao
E-mail : yaodi@ict.ac.cn
WeChat: yaodi833