If $\widetilde{\mathsf{RU}}(\mathsf{W})$ is close to 1, it implies the induced distribution is uniform. In other words, nothing is interesting that stand out, hence all the features in W vector have close to similar values. On the other hand if $\widetilde{\mathsf{RU}}(\mathsf{W})$ is less than some threshold, say β, there are high chances of certain distinguishable features. We are interested in the latter case as they are significantly different from others. In order to cull such significant features, we use a second parameter α that acts as a "cut-off" threshold to decide if an element in W is significant. All elements that satisfy the α threshold are removed from W and

Algorithm 1 Culling Significant Features

1: **Input:** Sub-matrix m^c of size $n_c \times D$ of cluster c
2: **Parameters:** $\alpha := \alpha_0; \beta := \beta_0; S_c := \emptyset$
3: **Initialization:** $S_c := \emptyset; k := 0;$
4: Compute weight vector W;
5: Compute $\theta := \widetilde{\mathsf{RU}}(\mathsf{W})$;
6: **while** $\theta \leq \beta$ **do**
7: $\quad \alpha = \alpha \times 2^{-k}; k++;$
8: $\quad$ **for each** $w_i \in \mathsf{W}$ **do**
9: $\quad\quad$ **if** $w_i \geq \alpha$ **then**
10: $\quad\quad\quad S_c := S_c \cup w_i; \mathsf{W} := \mathsf{W} - w_i;$
11: $\quad$ Update $\theta := \widetilde{\mathsf{RU}}(\mathsf{W})$
12: **Output:** S_c

put in S (i.e. significant feature set). We then perform the second iteration with the new W vector and make parameter α *slightly* relaxed. We keep repeating the process till the relative uncertainty exceeds β. Once the loop terminates, S contains the significant feature-set of cluster c. For the complete pseudo-code of our approach, see Algorithm 1. β parameter decides the threshold of relative uncertainty. Setting β to lower value would cause the significant sets to be larger compared to setting it to higher values. In general, value of β parameter for our case studies were set between 0.87 and 0.95. Initial value of α depends on the initial distribution of W. Based on our experience, a good starting point would be to set it to $min[\mathsf{W}]$. Line 7 in Algorithm 1 indicates the *decreasing factor* (or relaxing factor) of α that one may have to tune for subsequent iterations. Value of α and the decreasing factor assert a trade-off between faster run time versus better results. Figures 10a and 10b illustrate the results of applying our algorithm over sub-matrices of cluster 8 and 18, respectively. The portion to the left of every vertical blue line indicates the number of features that were part of the significant set (for intermediate iterations). The red line however indicates the iteration when our algorithm terminated. It is also evident from our results that the right portion of red line (which were not part of the significant set) are close to being uniform with not just very low weights but are also nearly indistinguishable. Hence, such features are deemed unimportant.

4 *EPIC* Framework Evaluation

Before we apply EPIC framework to real-world *geoMobile* datasets, we first briefly evaluate and compare the performance of our framework with other existing methods.

Since our framework consists of multiple components such as clustering and visualization, we compare each of these components individually with state-of-the-art baselines. The evaluation is conducted from two perspectives: performance of i) LE+DBSCAN clustering performance in comparison with other major clustering algorithms, and 2) Lt-SNE visualization algorithm based on local space contraction property.

4.1 Clustering Performance

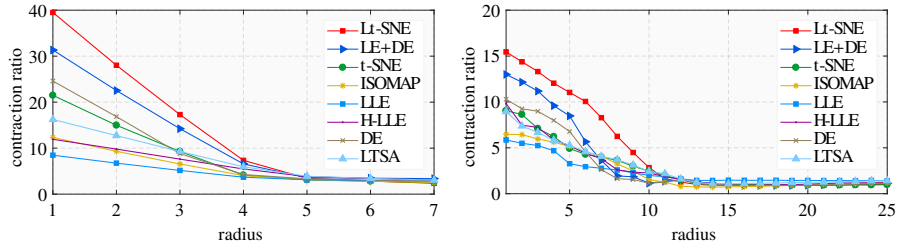
Table 1 shows the performance of clustering on case study 1's data matrix with respect to different algorithms under three clustering evaluation criteria: CalinskiHarabasz, Silhouette, DaviesBouldin – ↑ indicates higher values show better performance, similarly ↓ indicates otherwise. LE+DBSCAN's clustering performance dominates on two evaluation criteria and is reasonably good on the other one, showing the efficacy of our clustering approach. Similar results are also hold for dataset 2.

Table 1. Clustering Performance Evaluation on dataset 1 (see § 5.1).

Algo./Eval.	**CalinskiHarabasz ↑**	**Silhouette ↑**	**DaviesBouldin ↓**
Kmeans++	31.4021	0.1494	1.2768
Agglomerative	39.0319	0.3692	1.0714
Bi-Clustering	29.6709	0.4036	3.9815
LE+DBSCAN	274.132	**0.7082**	**0.5998**
DE+DBSCAN	**305.5648**	0.6899	0.6111
LLE+DBSCAN	25.4858	0.4413	1.0377
ISOMAP+DBSCAN	144.9113	0.5642	0.7481
LTSA+DBSCAN	116.2911	0.6081	0.9301

4.2 Best Local Space Contraction Property

Local Space Contraction Property: We provide justification for adopting LE approach in conjunction with t-SNE by analyzing the *local space contraction* effects in the (low dimensional) latent feature space. Building upon the work of [Rifai et al(2011)Rifai, Vincent, Muller, Glorot, and Bengio], we define the *contraction ratio* as the ratio of distance between two points in the input space and the distance mapped in the low dimensional feature space. Contraction ratio helps illustrate the deformation of the latent feature space in local regions. To measure this isometric property, we compute the average distance ratio of a point x randomly generated on a sphere of radius r centered at a fixed point x_0 in the input space over its corresponding distance in the feature space as a function of r. This function yields a curve called the *contraction curve*.



a. Contraction curves (WINE). b. Contraction curves (MNIST).

Fig. 11. Contraction curve of Lt-SNE have highest contraction ratio on two benchmark datasets.

Best Results: We compute the contraction curves on two benchmark datasets: WINE-dataset[2] and MNIST-dataset[3] and compare them (see Figure 11) with other major dimension reduction techniques: Deep-Autoencoder (DE), Local Linear Embedding (LLE), Local Tangent Space Alignment (LTSA), ISOMAP, Hessian LLE (H-LLE). These results also holds for other contraction curves such as Maximum Variance Unfolding (MVU), Kernel PCA and Probabilistic PCA which we do not show here. The process for generating contraction curves is as follows: x_0 is picked at centroid of a random cluster (using class label information) in the input space in order to study the propagation effects of contraction ratio from the center of a cluster. A random point x is generated at distance r on a sphere centered at x_0 and is included in the dataset. On this appended dataset, we apply dimension reduction techniques. In this process, we first reduce the input dimensions of the data (13 for WINE and 784 for MNIST datasets) to its intrinsic dimensions (3 and 12, respectively). Next, we apply t-SNE to embed the intrinsic data into a two dimensional space. An alternative and less expensive way to compute contraction curves is to implement *"out of sample extension"* methods [Bengio et al(2004)Bengio, Paiement, Vincent et al].

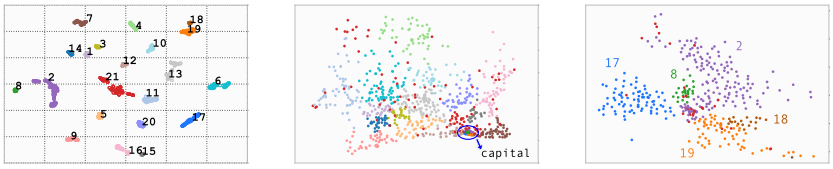
Figure 11 shows the contraction curves for major dimension reduction techniques. For both the datasets, Lt-SNE produces the *highest contraction ratio*, yielding low-dimensional maps with tighter clusters. Further, the strength of contraction gradually decreases with radius – until the effect vanishes marking the end of cluster radius size. Intuitively, Lt-SNE encourages contraction of neighborhood data points in the map since LE places data points on mutually orthogonal axes which, upon further applying t-SNE, helps produce tight clusters. Thus, Lt-SNE is capable of creating more distinguishable gaps between the clusters and in visualization. The same effect can be observed in the case of Deep-Autoencoder (with a depth of four layers), but with less contraction strength than Lt-SNE. On the other hand, t-SNE in one-shot reduction (i.e. directly reducing from the input dimension to $\mathbb{R}^2$) can produce a slightly lower contraction ratio than DE. Interestingly, applying LE in conjunction with DE (LE+DE) significantly boosts up the contraction ratio of DE but still remains lower than the Lt-SNE. This further indicates that the amalgam of LE and t-SNE are well suited to achieve high contraction ratio. Lastly, LLE produces the smallest ratio, suggesting that the resulting mapping contains more loose clusters as compared to others.

[2] `https://archive.ics.uci.edu/ml/datasets/Wine`
[3] `http://yann.lecun.com/exdb/mnist/`

Hidden Physics Behind the Dimension Reduction: A close analogy can be made between the contraction ratio and the strength of an electric field around a charged point. Just like the electric field strength propagates inversely proportional to square of radius, in the vein the strength of contraction ratio decays non-linearly as an inverse function of the radius. However, *unlike an electric field where the strength is equal in all directions, in the case of contraction ratio, the strength varies along the tangent space directions of the manifold on which the data is embedded non-linearly*. For instance, Figure 9b depicts the contour lines corresponding to the same level of contraction ratios in the low dimensional feature space. As expected, the shape of contour lines are not necessarily spherical but elongated along the dense regions of data points and falls off along the orthogonal direction which corresponds to the drop in the density of data points. These contraction curves reveal the internal feature transformation made by dimension reduction techniques along with their field strength (i.e., contraction ratio) & range (i.e., radius where contraction ratio becomes a constant). Such a comparison intuitively aids in the choice of the best dimension reduction technique in accordance with the application domain.

5 Case Studies

We primarily focus on analyzing two geoMobile datasets: i) a mobile call detail record (CDR) dataset collected from a nation-wide cellular network; and ii) a subway transit record dataset from a large city in China. The goal of this section is two-fold: 1) show the efficacy and generality of *EPIC* framework to wheel out interesting latent patterns from the datasets under multiple settings, 2) show tactical results and provide their interpretations. We share our experience in the form of three case studies.



a. Data points mapped over a low-dimensional space $\mathbb{R}^2$.

b. Cluster results overlaid over a geographic map of the nation.

c. Cluster results overlaid over a geographic map of the capital city & its close vicinities.

Fig. 12. Results showing 21 clusters (or distinct communication patterns) obtained from Dataset 1 (*best viewed in color*).

5.1 Case Study 1: Revealing Communication Patterns

In this case study, we use dataset 1 to extract communication patterns (driven by user actions such as making a voice call) between different origin and destination towers. Earlier in § 2, we observed a *locality effect* prevailing in communication patterns between towers in this dataset, suggesting people tend to call others more

often who are geographically closer to them. In this case study, we try to find such communication patterns, and to do so, start by representing this geoMobile dataset in the so-called form of *origin-destination* (or data) matrix. Mobile voice (and SMS) calls between users data in the cellular network as captured at cell tower levels are represented as OD matrices where origins are the cell towers calls originating from, destinations are cell towers these calls terminating at, and the entries in the OD matrix represent the number of calls between an origin-destination pair during some fixed time interval (in our case, average hourly calls). We formulate our problem using an input OD matrix of size $N \times D$, where the set of origins (or rows) and destinations (or columns) correspond to the set of unique towers, i.e. $N = D$. Each cell value x_{ij} in the square-OD matrix correspond to the average number of hourly calls made from the origin tower i to the destination tower j. Cell value x_{ij} will represent the number of *local* calls for tower i when $i = j$. In this data matrix, origins act as data points where as the destination towers act as the feature vector. Therefore, our clustering approach will group origin towers based on their call patters driven by human actions. In other words, two origin towers will be in the same cluster if both their outgoing call distributions to destinations towers are similar. It is important to note that the input data matrix has no information about the geographic coordinates or distances between towers. Results obtained by applying *EPIC* on this data matrix are shown in Figure 12a. There are a total of 21 well-separated clusters, representing 21 distinct communication patterns in the dataset. Using GPS coordinates, in Figure 12b we overlay the cellular-towers from these 21 clusters over a geographic map of the nation. Except for cluster ●21, all other clusters represent *regional* communication patterns of varying localities and sizes. Since these communication patterns are driven by human behavior, these distinct patterns capture social interactions in this African nation. We look more closely at the regional patterns in the capital city of the nation (see Figure 12c) as it comprises of over 300 out of the 900 towers of the nation. It is interesting to observe that the city itself is divided into five distinct communication "zones" driven by user interaction (in this case, call or message) and behavior: cluster ●2 which is the largest in the city, cluster ●8, cluster ●17, cluster ●18 and cluster ●19. Finally, the towers in cluster ●21 are sparsely distributed across the nation, most of which have relatively low overall call volumes and many are located along major transportation networks. This suggests that cluster ●21 represents call activities of users in transit across the nation. Although this approach is clustering origin towers, the same observations would hold true from the destination towers' perspective – this is because we observed that our OD data matrix is approximately symmetric.

In the context of this case study, each *significant set* of a cluster captures a particular kind of *human behavior*. In other words, each cluster's significant set are a set of features (or destination towers) that were most critical to that cluster's formation. Using the algorithm discussed earlier in § 3.3, we cull the significant sets for each of the 21 clusters, and visualize them in Figure 13 using a Venn diagram. Each circle (labeled using cluster number) in the Venn diagram represents a significant set of the corresponding cluster; size of the circle indicates the size of its significant set. Two circles intersect if their significant sets share common features (or destination towers). Metrics such as "jaccard similarity" can be used to *quantify* the similarity of human behavior among two intersecting significant sets. From Figure 13 and by

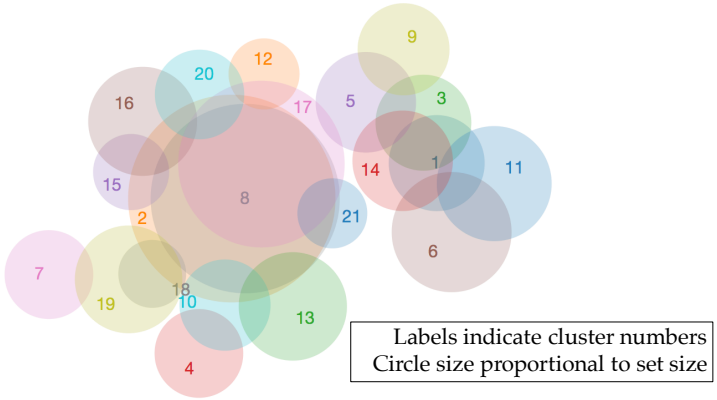
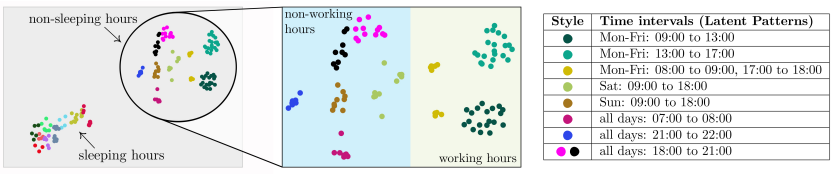


Fig. 13. Venn diagram of cluster-specific significant sets

further investigating the geographical features of the capital city, we find that cluster 2 (the mainland part of the capital city) not only has the largest significant set, but also intersects with a diverse set of other clusters. This suggests that the capital city is likely a hodgepodge of residents and a mobile population that originally come from other parts of the nation who still maintain strong social, commercial or other interactions with the rest of the nation. A similar (but to a less degree) pattern holds for cluster 1 which represents towers in a second-tier city in the nation. We see strongest similarities in communication patterns between clusters 18 and 19, as well as between clusters 2 and 8, reflecting their highly localized and close-knit communication patterns. Despite its towers distributed across the nation, cluster 21 intersects mostly with clusters 2, 8 and 17 representing towers in either the capital city or its suburbs, implying its communication pattern is due to users from the capital city and its vicinities travel across the nation.

5.2 Case Study 2: Temporal Communication Patterns

In this case study, we use the same dataset as in the previous case study to investigate if different hours of the day across the week have any similarity in their call patterns. For instance, do calling behaviors differ between morning and evening hours? How about weekends? Obtaining such insights would assist cellular operators to profile different hours (and days) based on user demands and usage to deploy, manage energy requirement and provide other personalized and value-added services.



(a) results in low-dimensional map (b) "latent" patterns as seen for *non-sleeping hours*

Fig. 14. Temporal similarities in communication patterns (Dataset 1)

To extract such latent patterns, we treat every data point to represent a day of the week and an hour. Therefore, we have 168 data points x_i for $i = 1, 2, .., 168$ (7 days of a week $\times$ 24 hours in a day). Every data point is represented by a N-feature vector f_j for $j = 1, 2, .., N$, where N is the number of towers. Each feature in the vector represents a tower and the value represents the *non-local* calls recorded by that tower, aggregated for the entire data set. In other words, given a data point $x_{d=MON,h=08}$ (i.e. day=Monday and hour=08:00 to 09:00 hours), each feature of the data point $x_{d=MON,h=08}$ would represent the average number of non-local calls recorded by that tower every Monday from 08:00 to 09:00 hours over a period of around 3 months. We now represent our data points and feature vectors as a data matrix X of size $168 \times N$ such that rows represent the data points and columns represent the N towers. Figure 14 shows the results by applying *EPIC* framework over X. In the $\mathbb{R}^2$ map (see Figure 14a), we clearly see two well-separated regions, one that captures data points representing usual *sleeping hours* (22:00 to 06:00) while the other represents *non-sleeping hours*. Looking more closely at the non-sleeping region, we observe some interesting patterns. The right half of this region seems to capture data points representing *working hours*, whereas the left half captures hours when people are at home. A complete list of the intuitively inferred "latent" patterns are listed in Figure 14b. We also see some outliers (anomalies) in the results indicating certain hours in the week have unique patterns. Although we considered hourly intervals for illustrating temporal communication similarities, one could also opt for intervals with smaller/higher temporal granularity. Intuitively, the extracted patterns suggest that the underlying reasoning behind the formation of clusters are related to human behavior, community interactions, social features, geographic features, etc. All in all, we show that *EPIC* framework is able to find some very interesting patterns in this case study.

5.3 Case Study 3: Temporal Variations in Human Mobility

In the third case study, we use Shenzhen Subway System's dataset (dataset 2) to gain insights about temporal variations in human mobility. As discussed earlier, we preprocessed the data to extract trip information. We also categorized users (as regular/adhoc) and their trips (as morning/evening/midday).

In order to investigate *"if"* and *"how" EPIC* is able to extract temporal variations in human mobility, we apply our framework to multiple data matrices, where each data matrix represents a particular time of the day (morning/evening/midday). We aggregated our processed dataset to obtain the *number of trips* between every pair of origin-destination subway station. We then build an OD matrix of size $N \times D$, such that every cell in the matrix represents the number of trips from the origin subway station to some destination subway station. As there are 118 unique subway stations in Shenzhen Metro, we have a matrix of size 118×118, i.e. $N = D = 118$. Labeling the records and users enable us to generate a number of OD matrices. For example, an OD matrix could represent trips made by *regular* users during *morning* hours. Note, rather than just looking at similarities between individual origin-destination pairs, our approach groups together origin data points based on the similarity in the distribution of the number of trips with all other destinations. Shenzhen Metro has 5

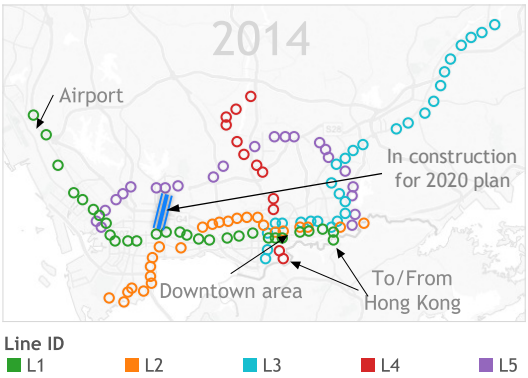
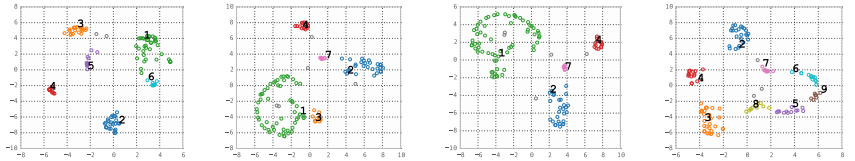


Fig. 15. Shenzhen Metro Subway Station and Line Map

subway lines (see Figure 15). If we assume that each subway line is independent of each other without the possibility for commuters to transfer from one line to another; applying our framework to such a dataset should ideally extract at least 5 clusters, where each cluster represents a particular subway line. This is because all possible pairs of origin-destination stations (representing a trip) are limited by the set of stations that are part of a particular subway line, owing to our assumption that people cannot transfer to other subway lines. Therefore, the probability of a commuter entering subway line A and exiting at subway line B is 0. Likewise, if the user enters and exits on subway stations that are part of the same subway line A, it is very likely that the probability of such trips will be greater than 0. However this assumption does not hold true for Shenzhen metro, as there are multiple subway stations that act as transfer points between different subway lines. But it is fair to assume that in the interest of reducing travel times, transit operators would design the subway lines so as to minimize transfers.



a. M_a: Regular user; Morning; b. M_b: Regular user; Evening; c. M_c: Regular user; Mid-day; d. M_d: Adhoc user; Mid-day;

Fig. 16. Output for four OD matrices (obtained from Dataset 2) categorized using temporal (morning/evening/mid-day) and user (regular/adhoc) labels. Numbers besides data points represent cluster dentifiers. Gray data points indicate outliers.

We construct four different OD matrices – M_a, M_b, M_c, and M_d. For instance, OD matrix M_a is built using trip information of regular users observed during the morning rush hours (construction details of other OD matrices is depicted in Table 2).

Figure 16 shows the results rendered by our framework in low-dimensional space, which are then further overlaid over Shenzhen's geographic map (see Figure 17). The first clear pattern we see is that certain clusters correspond to a particular subway line.

a. M_a: Regular user; Morning; b. M_b: Regular user; Evening; c. M_c: Regular users; Mid-day; d. M_d: Adhoc users; Mid-day;

Fig. 17. Results from Figure 16 mapped over Shenzhen's geographic map (obtained from Dataset 2).

	OD M_a	**OD** M_b	**OD** M_c	**OD** M_d
User label	Regular	Regular	Regular	Adhoc
Time interval	Morning	Evening	Mid-day	Mid-day
# of clusters	6	5	4	8
# of outliers	2	5	5	3
Pattern 1	✓	✓	✓	
Pattern 2	✓	✓	✓	✓
Pattern 3	✓	✓		✓
Pattern 4	✓	✓	✓	✓
Pattern 5	✓			✓
Pattern 6	✓			✓
Pattern 7		✓	✓	✓
Pattern 8				✓
Pattern 9				✓

Table 2. Comparison of Results

For example, all the red-colored clusters represent Subway Line 4 suggesting users traveling on this line have more localized traveling patterns who reach their destination with minimal transfers. Since this line also did not break into multiple clusters, it suggests that the trip volume distribution between any two subway stations are close to similar. A similar observation is observed with the dark blue clusters, which represents Subway Line 5. Therefore, by quick visual inspection of Figures 16 and 17, we are able to find *probable* patterns, which we list in Table 2. We consider a *"pattern"* to be a set of clusters, one from every OD-matrix if present. For example, we refer to one of our earlier observation regarding red clusters as "Pattern 4". Table 2 shows Pattern 4 is observed in all the OD matrices M_a, M_b, M_c, and M_d. From our earlier discussion regarding trip volumes, we observed large number of morning rush hour trips for regular users originate from suburban areas and end near Shenzhen's central (or downtown) region. Pattern 1 represents such trips for regular commuters (i.e. home $\rightarrow$ workplace trip). We also observe that the compactness of pattern 1 (green clusters) in M_a and M_b clearly differ. Note, this pattern represents majority of the trips during morning and evening rush hours. One plausible reason for this diversity in compactness could be that during the morning hours, all the trips end near the downtown area. On the other hand, evening trips appear to be dispersed. This could be due to the fact that while people travel back home, majority of the trips start from the downtown area but end at different regions around Shenzhen. Hence, we see an increased degree of spatial dispersity in the low-dimensional map for M_b in Figure 16b. Patterns seen during mid-day hours and evening hours for regular users seem to be almost the same. Even though the trip volumes during mid-day hours are significantly lesser than the rush hours, our approach is able to obtain the clusters. The interesting part in OD matrices

M_b and M_c is that the green cluster contains many subway stations from multiple Subway Lines L1, L2 and L5. This indicates all those subway stations have a higher degree of similarity in the travel patterns, thus suggesting dependency among each other. A possible side effect of such dependencies is increase in line transfers between subway lines. *This may be the reason why one of the future plans of Shenzhen metro is to establish a new track connecting Lines L1 with L5* (see Figure 15) [Metro(2015)]. For OD M_d representing adhoc users, we obtained relatively more number of clusters compared to regular users, where certain subway lines are partitioned. One probable reason could be that adhoc users (i.e. visitors, tourists, etc.) tend to take shorter trips within the central region of the city.

EPIC framework yields very interesting results for all the three case studies. Visualizing clusters in a low-dimensional ($\mathbb{R}^2$) map and further relating raw features to the cluster's formation adds different perspectives to interpret the clusters.

6 Conclusion

In this paper, we used the term geoMobile datasets to emphasize data that exhibit geo-spatial and human-behavioral features. To effectively handle high dimensional and skewed feature distributions inherent in geoMobile data, we developed *EPIC* framework to extract latent structures by combining and improving upon existing non-linear kernel clustering methods. We also uncover a theoretical reason for t-SNE's success and enhance it further to develop a visualization technique called Lt-SNE. In conjunction, we provide justifications on the effectiveness of our approach by studying & comparing contraction curves with other major dimension reductions techniques. Further, we developed a novel method to characterize the clusters based on raw features to aid in natural interpretation of the latent patterns. The tactical results obtained from our geoMobile datasets are very interesting. In this regards, our work yields an important tool in aiding data scientists to analyze diverse geoMobile datasets and uncover useful actionable knowledge embedded in them.

7 Acknowledgements

This research was supported in part by DoD ARO MURI Award W911NF-12-1-0385, DTRA grant HDTRA1- 14-1-0040, NSF grant CNS-1411636, CNS-1618339 and CNS-1617729.

References

Alsheikh et al(2016)Alsheikh, Niyato, Lin, p. Tan, and Han. Alsheikh MA, Niyato D, Lin S, p Tan H, Han Z (2016) Mobile big data analytics using deep learning and apache spark. IEEE Network 30(3):22–29, DOI 10.1109/MNET.2016.7474340

Baratchi et al(2014)Baratchi, Meratnia, Havinga et al. Baratchi M, Meratnia N, Havinga PJM, et al (2014) A hierarchical hidden semi-markov model for modeling mobility data. In: ACM Ubicomp

Belkin and Niyogi(2003). Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. NIPS

Bengio et al(2004)Bengio, Paiement, Vincent et al. Bengio Y, Paiement JF, Vincent P, et al (2004) Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. NIPS
Fan et al(2014)Fan, Song, and Shibasaki. Fan Z, Song X, Shibasaki R (2014) Cityspectrum: A non-negative tensor factorization approach. In: ACM Ubicomp
Hristova et al(2016)Hristova, Williams, Musolesi et al. Hristova D, Williams MJ, Musolesi M, et al (2016) Measuring urban social diversity using interconnected geo-social networks. In: ACM WWW
Ihler and Smyth(2006). Ihler AT, Smyth P (2006) Learning time-intensity profiles of human activity using non-parametric bayesian models. In: NIPS
Kling and Pozdnoukhov(2012). Kling F, Pozdnoukhov A (2012) When a city tells a story: urban topic analysis. In: ACM SIGSPATIAL
Krishnamurthy(2011). Krishnamurthy A (2011) High-dimensional clustering with sparse gaussian mixture models. Unpublished paper pp 191–192
Lakhina et al(2004)Lakhina, Crovella, and Diot. Lakhina A, Crovella M, Diot C (2004) Diagnosing network-wide traffic anomalies. In: ACM SIGCOMM Computer Communication Review
Lv et al(2015)Lv, Duan, Kang, Li, and Wang. Lv Y, Duan Y, Kang W, Li Z, Wang FY (2015) Traffic flow prediction with big data: a deep learning approach. IEEE Transactions on Intelligent Transportation Systems 16(2):865–873
van der Maaten and Hinton(2008). van der Maaten L, Hinton G (2008) Visualizing data using t-sne. Journal of Machine Learning Research (JMLR)
Manor and Perona(2005). Manor L, Perona P (2005) Self-tuning spectral clustering. In: NIPS
Metro(2015). Metro S (2015) Subway construction plan. `http://www.szpl.gov.cn/main/zsgg/200707090211041.shtml`
Ng et al(2002)Ng, Jordan, and Weiss. Ng AY, Jordan MI, Weiss Y (2002) On spectral clustering: Analysis and an algorithm. In: NIPS
Ozakin et al(2011)Ozakin, II, and Gray. Ozakin A, II NV, Gray A (2011) Manifold learning theory and applications. CRC press
Pressley(2010). Pressley A (2010) Elementary Differential Geometry. Springer
Rifai et al(2011)Rifai, Vincent, Muller, Glorot, and Bengio. Rifai S, Vincent P, Muller X, Glorot X, Bengio Y (2011) Contractive auto-encoders: Explicit invariance during feature extraction. ICML
Robusto(1957). Robusto C (1957) The cosine-haversine formula. The American Mathematical Monthly 64(1):38–40
Schiebinger et al(2015)Schiebinger, Wainwright, Yu et al. Schiebinger G, Wainwright MJ, Yu B, et al (2015) The geometry of kernelized spectral clustering. The Annals of Statistics 43(2):819–846
Städler and Mukherjee(2013). Städler N, Mukherjee S (2013) Penalized estimation in high-dimensional hidden markov models with state-specific graphical models. Ann Appl Stat
Wallach et al(2009)Wallach, Mimno, and McCallum. Wallach HM, Mimno DM, McCallum A (2009) Rethinking lda: Why priors matter. In: NIPS
Wang et al(2012)Wang, Hu, Xu et al. Wang Z, Hu K, Xu K, et al (2012) Structural analysis of network traffic matrix via relaxed principal component pursuit. Comput Netw
Witayangkurn et al(2013)Witayangkurn, Horanont, Sekimoto et al. Witayangkurn A, Horanont T, Sekimoto Y, et al (2013) Anomalous event detection on large-scale GPS data from mobile phones using hidden markov model and cloud platform. In: ACM Ubicomp
Yuan et al(2012)Yuan, Zheng, and Xie. Yuan J, Zheng Y, Xie X (2012) Discovering regions of different functions in a city using human mobility and pois. In: ACM SIGKDD
Zhang et al(2014)Zhang, Huang, Li et al. Zhang D, Huang J, Li Y, et al (2014) Exploring human mobility with multi-source data at extremely large metropolitan scales. In: ACM MobiCom
Zhang et al(2013)Zhang, Wilkie, Zheng, and Xie. Zhang F, Wilkie D, Zheng Y, Xie X (2013) Sensing the pulse of urban refueling behavior. In: ACM Ubicomp
Zhang et al(2005)Zhang, Ge, Greenberg, and Roughan. Zhang Y, Ge Z, Greenberg A, Roughan M (2005) Network anomography. In: ACM SIGCOMM IMC

8 APPENDIX

8.1 Proof of Proposition 1

Proof: KDE is a non-parametric way to estimate probability density function; it leverages the chosen kernel in the input space for smooth estimation. Given sub-manifold

density estimates $p(\mathbf{y}_i)$ for data points $\mathbf{y}_i \in \mathbb{R}^d \ \forall i$, we want to find a representation $\mathbf{z}_i \in \mathbb{R}^p \ \forall i$, $p < d$, such that the new density estimates $q(\mathbf{z}_i)$ agrees with the original density estimates. Here K_H, K_L denote the kernel in higher, lower dimensions, h is the kernel bandwidth and N is the number of data points. KDE's in higher and lower dimensions (assuming bandwidth h remains the same) are given by:

$$p(\mathbf{y}) = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{h^d}K_H\ \frac{\|(\mathbf{y}-\mathbf{y}_j\|_d}{h}\Bigg)\Bigg(q(\mathbf{z}) = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{h^p}K_L\ \frac{\|\mathbf{z}-\mathbf{z}_j\|_p}{h}\Bigg)$$

such that $\int K(u)du = 1$. The objective function of KL divergence loss for KDE can be computed as follows:

$$\mathcal{L} = \min_{\mathbf{z}} KL(p\|q) = \min_{\mathbf{z}} \sum_{i=1}^{N}\Bigg(p(\mathbf{y}_i)\log\frac{p(\mathbf{y}_i)}{q(\mathbf{z}_i)}$$

$$= \min_{\mathbf{z}} \frac{1}{Nh^d}\sum_{i=1}^{N}\sum_{j}\Bigg(K_H(\mathbf{y}_i,\mathbf{y}_j)\log\frac{\sum_j K_H(\mathbf{y}_i,\mathbf{y}_j)}{\sum_j\Bigg(K_L(\mathbf{z}_i,\mathbf{z}_j)} + c_1$$

Using log-sum inequality, we can show that,

$$\leq \frac{1}{Nh^d}\ \min_{\mathbf{z}}\ \sum_{i=1}^{N}\sum_{j=1}^{N}K_H(\mathbf{y}_i,\mathbf{y}_j)\log\frac{K_H(\mathbf{y}_i,\mathbf{y}_j)}{K_L(\mathbf{z}_i,\mathbf{z}_j)} + c_1$$

$$\leq c_2 \times \mathcal{J}\Bigg(+ c_1 \qquad \mathcal{J}\Bigg(\qquad\qquad \Bigg($$

$\mathcal{J}$ is the objective function of t-SNE (with specific kernels) which upper bounds (with a multiplicative scale and an additive constant) the estimated kernel density estimation loss function.

8.2 Proof of Proposition 2

Schiebinger et.al. [Schiebinger et al(2015)Schiebinger, Wainwright, Yu et al] studied normalized Laplacian embedding for i.i.d. samples generated from a finite mixture of nonparametric distribution. When the distribution overlap is small and samples are large, then with high probability they showed that the embedded samples forms a orthogonal cone data structure (OCS). Figure 18 shows that $(1-\alpha)$ fraction of two clusters are accumulated in a cone form of θ angle around e_1 and e_2 orthogonal axis.

Theorem 8.1 (Finite-sample angular structure) *There are numbers $b, b_0, b_1, b_2, \delta, t$ satisfying certain conditions such that the embedded data set $\{\phi(X_i), Z_i\}_{i=1}^n$ has $(\alpha, \theta)-$ OCS with*

$$|cos\theta| \leq \frac{b_0\sqrt{\varphi_n(\delta)}}{w_{min}^3 t - b_0\sqrt{\Bigg(\varphi_n(\delta)}}, \alpha \leq \frac{b_1}{w_m in^{1.5}}\varphi_n(\delta) + \psi(2t) \tag{11}$$

and holds with probability at least $1 - 8K^2\exp(\frac{b_2 n\delta^4}{\delta^2 + S_{max}(\overline{\mathbb{P}}) + B(\overline{\mathbb{P}})})$.

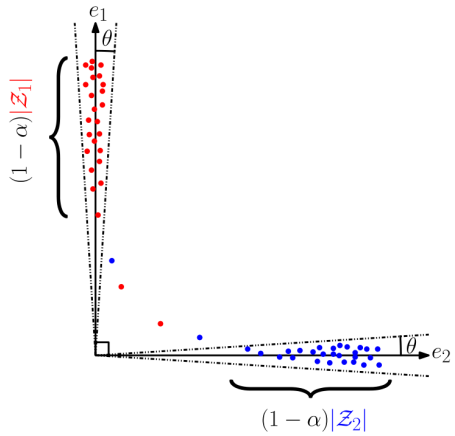


Fig. 18. Visualizing (α, θ)-OCS [Schiebinger et al(2015)Schiebinger, Wainwright, Yu et al].

Proof of Proposition 2: Our strategy is to exploit the OCS structure of the input data. Let $X \in \mathbf{R}^{N\times D}$ be the normalized data with unit norm, corresponding to p_{ij}, q_{ij} as higher, lower dimensional kernel densities and Z_1, Z_2 as normalization constant respectively. Let $X^{'} \in \mathbf{R}^{N\times d}, d < D$, be the normalized data obtained after LE dimension reduction and have similar corresponding variables $p^{'}_{ij}, q^{'}_{ij}, Z^{'}_1, Z^{'}_2$. Let $\beta \in (0, \frac{\pi}{2})$ and $\beta^{'}$ are angles between input feature vectors $< \mathbf{x}_i, \mathbf{x}_j >$ and $< \mathbf{x}^{'}_i, \mathbf{x}^{'}_j >$ respectively. Constant are denoted by $c_1, c^{'}_1, c_2, c_3, a_1, a_2 \geq 0$. Also σ and $\sigma^{'}$ are the kernel bandwidth of the estimated kernel densities in the X and X' input data respectively. Let i^{th} cluster has N_i samples out of K clusters. For our analysis, we will focus on this i^{th} cluster.

Since t-SNE preserves kernel density in lower dimensions, we will have $p_{ij} = q_{ij}$ and $p^{'}_{ij} = q^{'}_{ij}$. Some t-SNE related expressions that we will use for the proof are as follows,

$$p_{ij} = \frac{\exp\Big(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\sigma^2}\Big)}{\sum_{k\neq l}\exp\Big(-\frac{\|\mathbf{x}_k-\mathbf{x}_l\|^2}{2\sigma^2}\Big)}; q_{ij} = \frac{(1+\|\mathbf{y}_i-\mathbf{y}_j\|^2)^{-1}}{\sum_{k\neq l}(1+\|\mathbf{y}_k-\mathbf{y}_l\|^2)^{-1}}$$
$$Z_1 = \sum_{k\neq l}\exp\Big(-\frac{\|\mathbf{x}_k-\mathbf{x}_l\|^2}{2\sigma^2}\Big); Z_2 = \sum_{k\neq l}(1+\|\mathbf{y}_k-\mathbf{y}_l\|^2)^{-1}$$

Similar expressions can be obtained for $p^{'}_{ij}, q^{'}_{ij}, Z^{'}_1$ and $Z^{'}_2$. From these equations, we can show that,

$$\frac{(1-\cos\beta)}{\sigma_i^2} = \log\Big(\frac{1}{p_{ij}}-2\Big)\Big(\frac{1}{\log\sum_{k\neq l;k,l\neq i,j}\exp\Big(-\frac{\|\mathbf{x}_k-\mathbf{x}_l\|^2}{2\sigma^2}\Big)}\Big) \tag{12}$$

$$\|\mathbf{y}_i-\mathbf{y}_j\|^2 = \Big(\frac{1}{q_{ij}}-2\Big)\Big(\frac{1}{\sum_{k\neq l;k,l\neq i,j}(1+\|\mathbf{y}_k-\mathbf{y}_l\|^2)^{-1}}-1\Big) \tag{13}$$

Let, $c_1 = \sum_{k\neq l;k,l\neq i,j}(1+\|\mathbf{y}_k-\mathbf{y}_l\|^2)^{-1}$. Now according to Theorem 7.1, β' is bounded in $(\frac{\pi}{2}-2\theta, \frac{\pi}{2}+2\theta)$ with high probability, if (i, j) belongs to different class labels. In

general for small θ, we can assume that the different clusters form a separation angle (with respect to origin) such that $\frac{\pi}{2} - 2\theta > \beta$ i.e $\beta^{'} \geq \beta$ for all pairs of (i, j). Then according to Eq. 12, $p_{ij} \geq p_{ij}^{'}$ and therefore $q_{ij} \geq q_{ij}^{'}$, if (i, j) belongs to different class labels. Eq. 13 further yields,

$$\log \frac{c_1^{'}(1 + \|\mathbf{y}_i^{'} - \mathbf{y}_j^{'}\|^2) + 2}{c_1(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2) + 2} = \log \frac{q_{ij}}{q_{ij}^{'}} = \log \frac{p_{ij}}{p_{ij}^{'}} \geq 0$$

This shows that Lt-SNE always provide better mapping than t-SNE for $c_1^{'} \leq c_1$ which is generally the case. For small θ, we expect $p_{kl} > p_{kl}^{'} (\Rightarrow q_{kl} > q_{kl}^{'})$ for (k, l) belonging to different class and $p_{kl} \approx p_{kl}^{'} (\Rightarrow q_{kl} \approx q_{kl}^{'})$ for (k, l) belonging to the same class. This leads to $c_1^{'} \leq c_1$ since $q_{kl} \propto (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}$. Next, we establish an lower bound on this mapping ratio using this expression,

$$\log \frac{c_1^{'}(1 + \|\mathbf{y}_i^{'} - \mathbf{y}_j^{'}\|^2) + 2}{c_1(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2) + 2} = \frac{1 - \cos\beta^{'}}{\sigma^{'2}} + \log \frac{Z_1^{'}}{Z_1} + \frac{\cos\beta - 1}{\sigma^2} \tag{14}$$

For fixed β, $(\frac{\cos\beta - 1}{\sigma^2} - \log Z_1)$ term is constant. Now normalization constant $Z_1^{'}$ is the sum of kernel densities between samples within cluster itself and across other clusters. From Theorem 7.1, we know that $(1 - \alpha)$ fraction of a cluster belongs to a orthogonal cone structure with $\theta \in (0, \frac{\pi}{4})$ angle with high probability. Ignoring α samples (which add positive values to $Z_1^{'}$), we can provide a lower bound on $Z_1^{'}$ with the *same probability* bound as given in Theorem 7.1 for θ, α.

$$Z_1^{'} \geq \sum_{k=1}^{K}\Big((1-\alpha)^2 N_K(N_k - 1)e^{-\frac{(1-\cos 2\theta)}{\sigma^{'2}}} + \sum_{k\neq l}\Big((1-\alpha)^2 N_k N_l e^{-\frac{(1+\sin 2\theta)}{\sigma^{'2}}}$$

$$\underbrace{\Big(}_{\text{sum of densities within clusters}} \qquad \underbrace{\Big(}_{\text{sum of densities across clusters}}$$

$$Z_1^{'} \geq \frac{(1-\alpha)^2}{e^{\frac{(1-\cos 2\theta)}{\sigma^{'2}}}}\Big(\sum_{k=1}^{K} N_K(N_k - 1) + \sum_{k\neq l}\Big((1-\alpha)^2 N_k N_l e^{-\sqrt{2}\cos(\frac{\pi}{4} - 2\theta)}\Big)\Big)\Big($$

Finally, we can plug $Z_1^{'}$ in Eq. 14 and putting $\beta^{'} = \frac{\pi}{2} - \theta$ for getting lower bound, we obtain our final expressions.

$$c_2 = \frac{\cos\beta - 1}{\sigma^2} - \log Z_1 + \log\Big(\sum_{k=1}^{K} N_K(N_k - 1) + \sum_{k\neq l}\Big((1-\alpha)^2 N_k N_l e^{-\sqrt{2}}\Big)$$

$$\log \frac{c_1^{'}(1 + \|\mathbf{y}_i^{'} - \mathbf{y}_j^{'}\|^2) + 2}{c_1(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2) + 2} \geq \frac{\sqrt{2}\sin(\frac{\pi}{4} - 2\theta)}{\sigma_r^2} + 2\log(1-\alpha) + c_2$$

$$\implies \|\mathbf{y}_i^{'} - \mathbf{y}_j^{'}\|^2 \geq a_1\|\mathbf{y}_i - \mathbf{y}_j\|^2 + a_2$$

Here, $c_3 = \exp(\frac{\sqrt{2}\sin(\frac{\pi}{4} - 2\theta)}{\sigma_r^2} + 2\log(1-\alpha) + c_2) \geq 1$, $a_2 = \frac{c_3 + c_3 c_1 - c_1^{'} - 1}{c_1^{'}} \geq 0$ and $a_1 = \frac{c_3 c_1}{c_1^{'}} \geq 1$, if $c_1 \geq c_1^{'}$ which is the case for small θ. *This completes the full proof.*