Station-to-User Transfer Learning: Towards Explainable User Clustering Through Latent Trip Signatures Using Tidal-Regularized Non-Negative Matrix Factorization

Liming Zhang
George Mason University
lzhang22@gmu.edu

Dieter Pfoser
George Mason University
dpfoser@gmu.edu

Andreas Züfle
George Mason University
azufle@gmu.edu

ABSTRACT

Urban areas provide us with a treasure trove of available data capturing almost every aspect of a population's life. This work focuses on mobility data and how it will help improve our understanding of urban mobility patterns. Readily available and sizable farecard data captures trips in a public transportation network. However, such data typically lacks temporal signatures and as such the task of inferring trip semantics, station function, and user clustering is quite challenging. While existing approaches either focus on station-level or user-level signals only, we propose a Station-to-User (S2U) transfer learning framework, which augments user-level learning with shared temporal patterns learned from station-level signals. Our framework is based on a novel, so-called "Tidal-Regularized Non-negative Matrix Factorization" method, which incorporates a-priori tidal traffic patterns in generic Non-negative Matrix Factorization. To evaluate our model performance, a user clustering stability test based on the classical Rand Index is introduced as a metric to benchmark different unsupervised learning models. Using this metric, quantitative evaluations on three real-world datasets show that S2U outperforms two baselines methods by 7 – 21%. We also provide a qualitative analysis of the user clustering and station functions for the Washington D.C. metro and show how S2U can support spatiotemporal urban analytics.

CCS CONCEPTS

• **Computing methodologies** → **Regularization**; • **Applied computing** → **Transportation**.

KEYWORDS

Urban Mobility, Matrix Factorization, Temporal Signatures, Spatial-temporal Analysis.

ACM Reference Format:
Liming Zhang, Dieter Pfoser, and Andreas Züfle. 2020. Station-to-User Transfer Learning: Towards Explainable User Clustering Through Latent Trip Signatures Using Tidal-Regularized Non-Negative Matrix Factorization. In *28th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '20), November 3–6, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3397536.3422250

SIGSPATIAL '20, November 3–6, 2020, Seattle, WA, USA

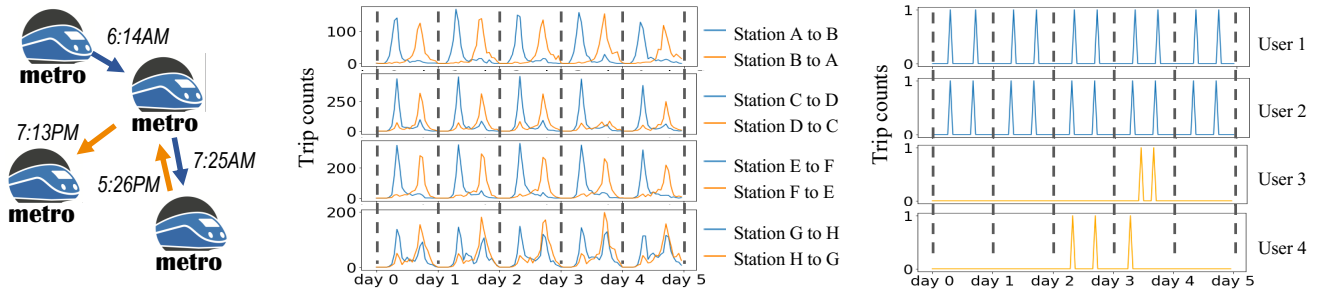
ACM ISBN 978-1-4503-8019-5/20/11…$15.00
https://doi.org/10.1145/3397536.3422250

1 INTRODUCTION

With two thirds of the population living in urban areas by 2050 [21], our future is characterized by mega cities in which urban mobility becomes a critical concern. For example, according to the 2019 INRIX Traffic Scorecard [25], people in large cities across the world waste an average of more than 150 hours stuck in traffic per year, wasting hundreds of billions of USD and creating unnecessary greenhouse gas emissions. For these reasons, many recent studies have focused on modeling and predicting human mobility in urban areas (cf. [29]). Paramount to improving urban mobility is to understand why people travel. Directly collecting trip purpose data through travel surveys is a long-standing and time intensive practice [20]. Mobile computing and crowdsourcing data [8–10, 31] provides us with new means to collect such information. Yet, a data driven approach is challenging, as available trip information does not typically capture trip semantics such as the trip purpose.

The goal of this work is to provide a machine learning framework to infer the trip purpose from, for example, commonly available farecard data from public transportation systems such as the metro. Each record represents a trip and is of the form <Card ID, Entry Station, Arrival Station, Entry Time, and Arrival Time>. Assuming that how people utilize the metro in urban areas reflects some underlying behavior, it is possible to utilize machine learning to discover common threads and patterns with respective to user behavior. For example, Figure 1(a) shows the daily trip of a single user (identified by Card ID). Each arrow is a trip between stations with its associated timestamp. There is only one timestamp for a trip because the entry or exit time are not always available in different farecard systems and most schedules of metro systems run so tight that duration of trips varies little and with little interest in our problem. In our toy example, the user enters the first station at $6:14am$, after a stop enters another station at $7:25am$ to go to the third station, which likely brings him to his work place. In the afternoon, another trip happens at $5:26pm$ going back to the second station. The user might have dinner there and then travels at $7:13pm$ to the fourth station, which could be the airport to catch a flight. With many such user trips, we can aggregate and group these trips by their origin-destination (OD) pairs in Figure 1(b), and considering the temporal trip distribution of users in Figure 1(c) we can make the following observations.

(1) **Tidal Patterns:** Using four origin-destination pairs of the Washington D.C. metro example in Figure 1(b), we observe morning peaks in the direction of $A \rightarrow B$, commonly paired with afternoon peaks in the opposite direction $B \rightarrow A$. This phenomenon is referred to as a *tidal traffic pattern* (cf. [1, 28]).



(a) Metro Farecard data (b) Directed traffic flows (c) User trip series

Figure 1: (a) A metro trip example - each arrow indicates a trip and timestamp; (b) four station pairs and their symmetric forwards (blue lines) and backwards (orange lines) passenger flows known as "tidal traffic"; (c) trip flows of two frequent travellers (Users 1&2 - blue lines) show a strong recurrent commuting pattern, while Users 3&4 have a pattern that is harder to predict.

(2) **Flow Outliers:** The second and fourth row in Figure 1 (b) show station pairs with non-symmetric patterns. For example, on Day 4, the afternoon return-flow from Station D to C is much lower than what we expected from the respective morning peak. Stations G and H have additional spikes throughout the day, which may be due to non-symmetric activities such as tourism or special events that are not observable in farecard data.

(3) **Commuters:** Many metro users are commuters, such as the users corresponding to the top two rows in Figure 1 (c), have a morning/afternoon trip-return trip pattern.

(4) **User Outliers:** In Figure 1 (c), Users 3 and 4 have irregular trips. On Day 4, User 4 has only a single trip without a return trip. Inferring trip purpose or typology of such users is challenging.

Based on these observations, we can abstract and highlight a number of unique challenges for inferring the trip purpose from farecard data. (1) The function of a metro station is user dependent, as the home station of one user may be the work station of another, and the "third place"[22] or the recreation station of yet another user. (2) Matching farecards to users is non-trivial, as one user can have multiple cards and one card can be used by different users. (3) Users making irregular trips are hard to categorize and make it difficulty to infer a trip purpose. These challenges blur the signal of each individual user and make inferring the trip purpose through a straightforward application of machine learning challenging and less convincing. To address these challenges, we propose a novel "Station-to-User (S2U)" transfer learning framework along with a domain-specific "Tidal-Regularized Non-negative Matrix Factorization (TR-NMF)" machine learning algorithm. This framework defines similarities of users by mapping them to a latent feature space learned from stations. Creation of this feature space exploits knowledge about "tidal" behavior of users having recurrent morning and afternoon peaks [3]. We also propose a clustering stability test as a cross-model evaluation metric to promote future benchmarking for station and user clustering research.

The remainder of this paper is organized as follows: After surveying the related work in Section 2, we introduce the datasets and formalize the problem of explainable user clustering in Section 3. Section 4 introduces our new S2U transfer learning framework with its novel TR-NMF ML model to achieve explainable clustering based on trip semantics. Section 5 provides a quantitative and qualitative evaluation of the proposed approach. Finally, Section 6 gives conclusions and provides directions for future research.

2 RELATED WORK

Early works on metro farecard data focus on descriptive statistics to characterize tidal patterns and dominant stations [10, 17]. Solutions have been proposed to infer the function of regions of a city based on individual mobility data in [36]. This work uses topic modeling to map point of interest and user visits of a region to latent topics. The latent topics that are leveraged to assess similarity between regions. Following this approach, it has been shown in [37] that the function of a region changes over time and that it is paramount to consider temporal dynamics. Specifically using farecard data, latent factor based solutions can recognize daily patterns for weekdays, weekends, and holidays [35].

Related to our approach, a recent matrix factorization based approach to infer the temporal functions of regions (or stations) has been proposed in [32]. This approach has been leveraged to identify tidal patterns of human mobility in [28]. What these efforts have in common is to infer the function of regions or stations. Our goal is to go a step further and to identify the "function" or signature of individual users to assess the similarity of users and to cluster them into groups of similar behavior. The work in [8] uses trajectory data and stop points to infer user-specific activities. However, using only origin, destination, and time information available in farecard data, it is not possible to infer stops at specific points of interest to directly infer the purpose of a trip.

Non-negative Matrix Factorization (NMF) based solutions have also been proposed for other problems related to urban mobility, such as predicting road traffic [11, 33] and predicting metro traffic demand [6]. These works provide powerful solutions to predict traffic, but lack explanatory power. To capture spatial and temporal mobility patterns, existing efforts [10, 23, 27] use NMF to explain temporal patterns in daily life, such as commuting pattern that concentrates on mornings and afternoons, and explains the function of urban areas. In a recent work [30], a context-aware tensor decomposition is used to explain urban mobility over space and time using a tensor factorization approach. These works model similar spatial and temporal urban dynamics, such as days having similar mobility patterns and regions having similar function. However, they do not for a similarity assessment and categorization of users and passengers. In contrast, our approach unwraps the signatures of individual metro users and to cluster them so to explain individual users and the purpose of their trips.

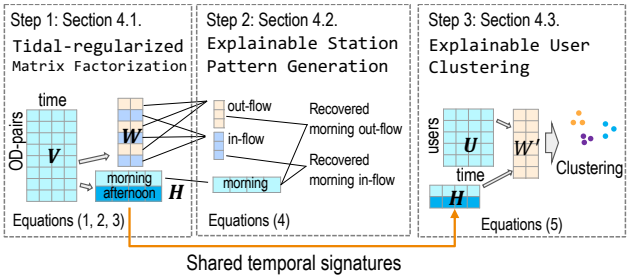


Figure 2: Station-to-User (S2U) Transfer Learning Framework

3 PROBLEM DEFINITION

This section formally defines a trip database and the data structures obtained from a trip database that will later be factorized to cluster stations and users.

Definition 3.1 (Trip Database). Let $\mathcal{U}$ be a set of metro users, let $\mathcal{S}$ be a set of metro stations, let $\mathcal{OD} = \mathcal{S} \times \mathcal{S}$ denote the set of all origin-destination station pairs, and let $\mathcal{T}$ be a set of time intervals or epochs. A trip database $\mathcal{DB}$ is a collection of tuples $(u, (o, d), t) \in \mathcal{U} \times \mathcal{S} \times \mathcal{S} \times \mathcal{T}$, where $u \in \mathcal{U}$ is a user, $(o, d) \in \mathcal{OD}$ is an OD-pair, $o \in \mathcal{S}$ is the origin station, $d \in \mathcal{S}$ is the destination station, $t \in \mathcal{T}$ is the start time of the trip.

Using the trip database, we can define a temporal flow matrix for all OD-pairsm which aggregates trip data and stores the number of trips, grouped by OD-pairs, for each time epoch.

Definition 3.2 (OD-pair Temporal Flow Matrix). An OD-pair temporal flow matrix is denoted as $\boldsymbol{V} \in \mathbb{R}^{|\mathcal{OD}| \times |\mathcal{T}|}$, such that:

$$\boldsymbol{V}_{(o,d) \in \mathcal{OD}, t \in \mathcal{T}} = |\{x \in \mathcal{DB} | x.o = o \wedge x.d = d \wedge x.t = t\}|$$

We further define a temporal flow matrix for each user that aggregates the number of trips per user grouped by time epochs independent of individual stations.

Definition 3.3 (User Temporal Flow Matrix). is denoted as $\boldsymbol{U} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{T}|}$, matrix such that:

$$\boldsymbol{U}_{u \in \mathcal{U}, t \in \mathcal{T}} = |\{x \in \mathcal{DB} | x.u = u \wedge x.t = t\}|$$

Given an OD-pair temporal flow matrix $\boldsymbol{V}$ and a user temporal flow matrix $\boldsymbol{U}$, our goal is to cluster users such that (latent) features extracted from temporal flow are maximized between users of the same cluster. Given these clusters, our goal is to evaluate these features of a cluster to explain the function of clusters and the latent semantic of users and their trips.

4 STATION-TO-USER (S2U) TRANSFER LEARNING FRAMEWORK

To better cluster users based on the purpose of their trips, we propose a framework to learn the temporal signature between stations and users in a transfer learning manner. The diagram of this Station-to-User (S2U) Learning Framework is shown in Figure 2 and has three main steps.

Step 1: Tidal-Regularized Matrix Factorization: Factorization of the OD-pair temporal flow matrix $\boldsymbol{V}$ (cf. Definition 3.2) to find latent temporal features $\boldsymbol{H}$ and latent trip features $\boldsymbol{W}$. To obtain interpretable features, we employ a tidal-regularized loss function to better fit the (empirically grounded) tidal pattern observed in urban mobility contexts ("commuters"). More details on this approach are provided in Section 4.1.

Step 2: Explainable Station Pattern Generation: Semantics-based aggregation of latent trip features $\boldsymbol{W}$ to reconstruct semantics-based inflows and outflows at each station. The reconstructed flows indicate, for example, the degree to which a station is a work destination (inflow) or a home destination (outflow). Details of this approach are given in Section 4.2.

Step 3: Explainable User Clustering: Mapping temporal flow of users $\boldsymbol{U}$ to the space spanned by shared temporal signatures $\boldsymbol{H}$ learned from decomposing the station flow matrix using transfer learning. This yields a user weight matrix $\boldsymbol{W'}$ containing the temporal features of each user. The reason for mapping users to the station space is that tidal features of stations are more stable and less noisy as shown in our experimental evaluation. This approach allows us to provide explainable behavioral differences between users, even if only a few observed trips available. More details on this step are found in Section 4.3.

4.1 Tidal-regularized Non-negative Matrix Factorization (TF-NMF)

We decompose matrix $\boldsymbol{V} \in \mathbb{R}^{|\mathcal{OD}| \times |\mathcal{T}|}$ into two non-negative matrices $\boldsymbol{W} \in \mathbb{R}^{|\mathcal{OD}| \times K}$ and $\boldsymbol{H} \in \mathbb{R}^{K \times |\mathcal{T}|}$, such that

$$\boldsymbol{V} \approx \boldsymbol{W}\boldsymbol{H} =: \hat{\boldsymbol{V}}, \tag{1}$$

where K is a positive integer, $\mathcal{OD}$ is the set of origin-destination pairs, and $\mathcal{T}$ is the set of temporal epochs (c.f. Definition 3.1). To find $\boldsymbol{W}$ and $\boldsymbol{H}$ we minimize a loss function $\mathcal{L}$ defined by the mean square approximation error and the l_1 and l_2 norms of $\boldsymbol{W}$ and $\boldsymbol{H}$ as follows [13]:

$$\begin{aligned} \mathcal{L}' = & \sum_i \sum_t (\boldsymbol{V}_{i,t} - \hat{\boldsymbol{V}}_{i,t})^2 \\ & + \alpha\eta(\|\boldsymbol{W}\|_1 + \|\boldsymbol{H}\|_1) + \alpha(1-\eta)(\|\boldsymbol{W}\|_2 + \|\boldsymbol{H}\|_2), \end{aligned} \tag{2}$$

where $\|\cdot\|_1$ is the l_1 norm of a matrix, $\|\cdot\|_2$ is l_2 (or Frobenius norm) of a matrix, and α, η are hyper-parameters.

Motivated by a tidal pattern observed in urban contexts such as for traffic and passenger volumes (cf. [1, 26, 28]), we observe that this pattern has strong temporal peaks, with the morning commute happening before $11am$, and a symmetric afternoon commute after $2pm$. We incorporate this *a-priori knowledge* into our NMF approach by adding **a tidal-regularized (TR) loss to the generic NMF loss function.** It acts as a soft regularization to guide learned temporal signatures towards a better fit to such a tidal pattern. To understand the tidal regularized loss, we partition factor matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ to separate tidal features corresponding to daily morning and afternoon peaks. This approach is illustrated in Figure 3 and described in the following.

(i) Grouping latent features by temporal semantics. Generic NMF does not consider (or understand) temporal ordering, as temporal epochs (columns in $\boldsymbol{U}$ and $\boldsymbol{V}$) are treated as nominal (but not ordinal) variables. As such, we sort latent features by their temporal semantics to understand and guide the learning process. We exploit that matrix $\boldsymbol{H}$ provides the temporal semantics of each latent feature. It describes each temporal epoch (such as each hour), by

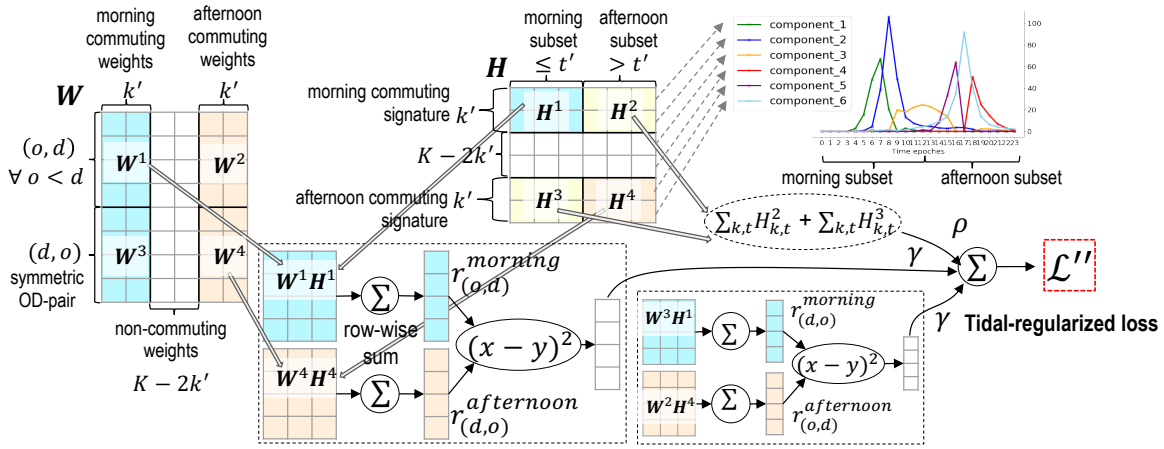


Figure 3: Partitions of matrix W and matrix H by Symmetric OD-Pairs and temporal ordering, and compositions of tidal-regularized loss

K latent features. Assuming tidal patterns, we expect some latent features to have larger weights for morning epochs, and some latent features to have larger weights in the afternoon partition. The example in the upper right corner of Figure 3 shows the temporal semantics of the Washington D.C. metro data obtained from NMF using $K = 6$ latent features. Latent features 1 and 2, have morning hour semantics. Features 5 and 6 relate to afternoon hours. Feature 3 relates to mid-day activities and Feature 4 captures late afternoons and evenings.

We swap lines in $\boldsymbol{H}$ such that the first $k' \leq K/2$ rows correspond to the morning features, the last k' rows correspond to the afternoon features, and the middle $K - 2k'$ rows capture all other (non-commuting) features. k' is a hyper-parameter of our model. To ensure that this swapping of columns in $\boldsymbol{H}$ does not affect the factor product $\boldsymbol{V}$, we perform the same swap orders among columns of $\boldsymbol{W}$ using the observation based on Lemma A.1 in Appendix A.

In the following, we will exploit this assumption to regularize the matrix factorization to leverage tidal patterns for a more accurate and more explainable factorization model.

(ii) Symmetric OD-Pairs: Next, we exploit the symmetry across origin-destination pairs in a sense that we expect an OD-pair (o, d) to have a flow symmetric to its inverse OD-Pair (d, o). This is a consequence of most passenger flow being generated by commuters. For example in Figure 1(b), for an OD pair that is mainly used in the morning, the symmetric OD pair will be mainly used in the afternoon. We drop all the OD-pairs having $o = d$, i.e., users entering and exiting the same station. A similar swap is done to make sure the upper half of rows in $\boldsymbol{W}$ meets the requirement of $o < d$ and the lower half of rows in $\boldsymbol{W}$ includes all OD-pairs with $o > d$ as illustrated in in the upper left of Figure 3.

With this notation, we will introduce different partitions for normal weight matrix $\boldsymbol{W}$ and temporal feature matrix $\boldsymbol{H}$ to allow semantics-based regularization.

(iii) Partitions of latent feature matrix H with subsets of latent features by temporal ordering. For each row of $\boldsymbol{H}$, we additionally divide the columns into two subsets: a *morning subset* and an *afternoon subset* (illustrated in the center of Figure 3). The columns of $\boldsymbol{H}$ represent time of day $t \in \mathcal{T}$ (such as each hour of a day), so the morning subset includes all hours before or equal to t' hour and the afternoon subset includes all hours after t' hour, where t' is a hyper-parameter. We simply select $t' = 12$ as noon by default. Finally, we get five partitions of $\boldsymbol{H}$ based on (i) and this subsetting: $\boldsymbol{H}^1, \boldsymbol{H}^2, \boldsymbol{H}^3, \boldsymbol{H}^4$, and the middle rows for non-commuting features. This partitioning ensures that the afternoon subsets of morning signatures $\boldsymbol{H}^2$ and the morning subsets of afternoon signatures $\boldsymbol{H}^3$ are strongly regularized to zero, or are directly set to zero during training. $\boldsymbol{H}^1$, $\boldsymbol{H}^4$ and the middle signatures are trained without any regularization. Such regularization is shown by the expression $\sum_{k,t} \boldsymbol{H}^2_{k,t} + \sum_{k,t} \boldsymbol{H}^3_{k,t}$ in the ellipsoid of Figure 3. It is the first part of our proposed Tidal-Regularized (TR) loss.

(vi) Temporal partitions of weight matrix W. Based on (i) and (ii), as a result of swapping rows and columns, we get different partitions for $\boldsymbol{W}$ and $\boldsymbol{H}$ as illustrated in Figure 3. For matrix $\boldsymbol{W}$, we subdivide the columns of $\boldsymbol{W}$ into three groups corresponding to k' morning features (left), k' afternoon features (right), and $K - 2k'$ non-commuting features (middle). The upper half of rows in $\boldsymbol{W}$ captures OD pairs with $o < d$, while the lower half is for OD pairs with $o > d$.

Components of Tidal-Regularized (TR) loss. We can now formalize the three components of TR loss (three arc lines pointing to $\sum$ sign before TR loss $\mathcal{L}''$ on the right side of Figure 3) as follows:

- Component 1 - zero-regularized $\boldsymbol{H}^2$ and $\boldsymbol{H}^3$;
- Component 2 - minimizing the differences between a OD-pair (o, d)'s total morning commuting flow and its symmetric OD pair's (d, o) afternoon commuting flow;
- Component 3 - minimizing the differences between the OD-pair (o, d)'s total afternoon commuting flow and its symmetric (d, o) morning commuting flow.

Next, we will explain each component in detail. For Component 1 (top arc), its goal is to create partitions $\boldsymbol{H}^2$ and $\boldsymbol{H}^3$ with zeros or close to zero because of our definition of temporal signatures. $\boldsymbol{H}^2$ are the afternoon subsets of morning commuting signatures, so $\boldsymbol{H}^1$ should be the part to learn the morning peaks of commuter flows. For the same reason, $\boldsymbol{H}^3$ are the morning subsets of afternoon commuting signatures, so $\boldsymbol{H}^4$ should contain the afternoon peaks of commuter flows. We use a zero-regularization to penalize any

element that is not zero, as follow:

$$\sum_{k,t} \boldsymbol{H}^2_{k,t} + \sum_{k,t} \boldsymbol{H}^3_{k,t}$$

For Component 2 (middle arc), we first reconstruct the matrix of all (o,d) morning trips as $\boldsymbol{W}^1\boldsymbol{H}^1$. Then, we accumulate all times t during the morning to get only one total flow for each (o,d) as

$$\boldsymbol{r}^{morning}_{(o,d)} = \sum_{t=1}^{t'} \left(\boldsymbol{W}^1_{(o,d),}\boldsymbol{H}^1_{,t}\right)$$

Similarly, afternoon commuting flows of symmetric OD pairs (d,o) are obtained by

$$\boldsymbol{r}^{afternoon}_{(d,o)} = \sum_{t=t'+1}^{T} \left(\boldsymbol{W}^4_{(d,o),}\boldsymbol{H}^4_{,t}\right)$$

The next step is to obtain, for an origin destination pair (o,d), the absolute differences of (o,d)'s morning flow $\boldsymbol{r}^{morning}_{(o,d)}$ and their symmetric OD pair (d,o) afternoon flows $\boldsymbol{r}^{afternoon}_{(d,o)}$. We use the squared difference $(\boldsymbol{r}^{morning}_{(o,d)} - \boldsymbol{r}^{afternoon}_{(d,o)})^2$ to penalize differences between morning peaks and symmetric afternoon peaks. This penalization helps to ensure that the matrix factorization is regularized to maintain the symmetric tidal behavior that we have observed empirically in Figure 1. Then, all squared differences for OD-pairs with $o < d$ are summed up as the 2^{nd} component. The squared differences are summed up and multiplied by a hyper-parameter γ as for the TR loss. If this component is substantial during training, the model will be penalized and be forced to optimize towards the direction of lowering such a penalty.

Component 3 (bottom arc) is computed analogously for the morning commute of (d,o)s as follows.

$$\boldsymbol{r}^{morning}_{(d,o)} = \sum_{t=1}^{t'} \left(\boldsymbol{W}^3_{(d,o),}\boldsymbol{H}^1_{,t}\right).$$

The afternoon commute of their symmetric OD-pairs (o,d) is calculated as

$$\boldsymbol{r}^{afternoon}_{(o,d)} = \sum_{t=t'+1}^{T} \left(\boldsymbol{W}^2_{(o,d),}\boldsymbol{H}^4_{,t}\right).$$

Then, all squared differences between symmetric OD-pairs are summed up and multiplied by γ and added to TR loss.

In summary, the Tidal-regularized loss is formulated by summing up all three components as follows:

$$\begin{aligned}\mathcal{L}'' = \gamma \sum_{d\le|\mathcal{S}|-1, o<d} \big((\boldsymbol{r}^{morning}_{(o,d)} - \boldsymbol{r}^{afternoon}_{(d,o)})^2 \\ + (\boldsymbol{r}^{morning}_{(d,o)} - \boldsymbol{r}^{afternoon}_{(o,d)})^2 \big) + \rho\big(\sum_{k,t} \boldsymbol{H}^2_{k,t} + \sum_{k,t} \boldsymbol{H}^3_{k,t} \big),\end{aligned} \tag{3}$$

where ρ, γ are hyper-parameters used to tune the algorithm.

Finally, the total loss $\mathcal{L}$ is obtained by summing up, both, the generic NMF loss and our newly-proposed TR loss as follows:

$$\mathcal{L} = \mathcal{L}' + \mathcal{L}''.$$

We use *Tensorflow*[1] and utilize the *Autodiff*[2] features, which automatically compute the gradient update rules to optimize $\boldsymbol{W}$ and $\boldsymbol{H}$. To terminate training, we either use a threshold for the reconstruction error $||\boldsymbol{V} - \hat{\boldsymbol{V}}||_1$ (which tells us the sum of absolute difference between raw and reconstructed flow matrix), or use a fixed number of training steps. Post training, we convert the latent representation to a unit vector (cf. [34]) and also update the corresponding normalized weight $\boldsymbol{W}$ as follows:

$$h_{k,t} = h_{k,t}/\sqrt{\sum_t^T h^2_{k,t}},\ w_{i,k} = w_{i,k}\sqrt{\sum_t^T h^2_{k,t}},$$

where $w_{i,k}$ is the element in i^{th} row and k^{th} column of $\boldsymbol{W}$, and $h_{k,t}$ is the element in k^{th} row and t^{th} column of $\boldsymbol{H}$. The complexity analysis of our algorithm is shown in Appendix B and potential early stopping to accelerate algorithm is shown in Appendix C.

4.2 Explainable station pattern generation based on latent temporal signatures

4.2.1 ***Relating temporal signatures to station functions:*** We reconstruct in- and outflows based on each semantic-based temporal signature group (defined in Part (i) of Section 4.1) for each station (cf. [4, 10]). For example, the morning inflow of a station means it attracts people for, e.g., work, and we refer to this as the "attractivity" of a station. Based on our previous arguments on TR loss, afternoon outflow for the same station indicates people leaving from work and it is symmetric to the morning inflow. A morning outflow of station generates people, e.g., homes and hotels. We refer to this as "generativity" function of a station. Different from existing works, we propose the novel TR Loss, which strongly guides learned temporal signatures to fit a-priori tidal patterns. The temporal signatures gain more explanatory power for different reconstructed in- and outflows of stations. Moreover, we can distinguish between different types of commuting (flexible work hours, etc.) for a station using each individual temporal signature (a row of $\boldsymbol{H}$) through a temporal signature's peak hour.

4.2.2 ***Semantics-based aggregation:*** Semantics-based aggregation is a procedure to get explainable station functions for each station, for example, Station $A \in \mathcal{S}$ in Figure 4. Aggregation is performed for each hour h of the day for which data is found in the corresponding column i in matrices $\boldsymbol{W}^1$ and $\boldsymbol{W}^3$. For example, for $h = 7am$ and $i = 1$, we select a morning subset of one specific temporal signature (a $7am$-peak temporal signature $\boldsymbol{H}^1_{1,\cdot}$ in Figure 4) according to the partitions of temporal signatures (defined in Part (iii) in Section 4.1). We also select from the column of partition $\boldsymbol{W}^1$ that corresponds to the selected hour of the day h (weights of the peaking signature at that time) for all the other OD-pairs $(A,\cdot)$ that originated from station A, like $(A,B), (A,C), (A,D)$, noted as $\boldsymbol{W}^1_{(A,\cdot),i}$. Then, we can reconstruct the outflow matrix for this specific temporal signature as:

$$\boldsymbol{V}^{A-out}_i = \boldsymbol{W}^1_{(A,\cdot),i}\boldsymbol{H}^1_{i,\cdot}$$

The total generativity $Gen_h(A)$ of station A at time h is computed as the sum of all elements in reconstructed flow matrix as follows:

$$Gen_h(A) = \sum_{i,t} V^{A-out}_{i,t}$$

For example, the $7am$-peak generativity of Station A is

$$Gen_{7am}(A) = \sum_{1,t} V^{A-out}_{1,t},$$

as column 1 of $\boldsymbol{W}^1$ corresponds to the $7am$ flow.

An analogous approach is used to compute the attractivity of Station A at time h. We first select h-peaking weights of all OD-pairs $(\cdot,A)$ destined for station A, like $(B,A), (C,A), (D,A)$, noted

[1] www.tensorflow.org

[2] https://www.tensorflow.org/tutorials/customization/autodiff

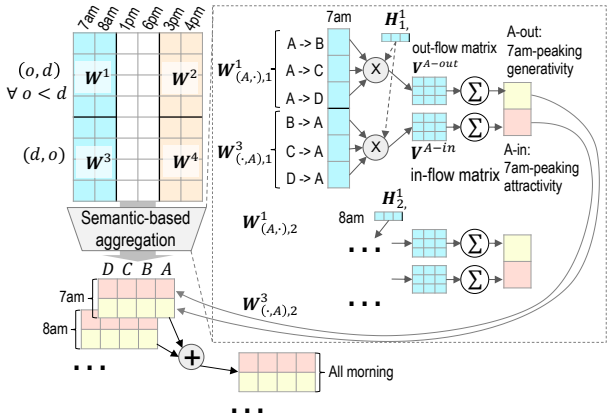


Figure 4: Semantics-based aggregation operation to get explainable generativity and attractivity station functions

as $\boldsymbol{W}^3_{(\cdot,A),i}$. Then, an inflow matrix $\boldsymbol{V}^{A-in} = \boldsymbol{W}^1_{(\cdot,A),i}\boldsymbol{H}^1_i$ is reconstructed. The total attractivity of station A at time h is defined as: $\sum_{i,t} \boldsymbol{V}^{A-in}_{i,j}$. The recovered in- and outflows inherit the semantic meaning of different stations. Stations with large early-hour in-flows are attractive places for work, where stations with large early-hour outflows are strong generative places such as residential areas. Section 5.3 gives examples for such a station function analysis in Washington D.C.

4.3 Using Temporal Station Signatures for Explainable User Clustering

To explain the semantics of users, we project their trips to the temporal signature space defined for stations in Section 4.1. In this section, we will first introduce how such transfer learning is conducted so that a stable and explainable user clustering can be found. Then, we introduce a novel Clustering Stability Test that can be used to universally judge the stability performance of methods with different internal procedures.

4.3.1 ***Transfer learning:*** Similar to general transfer learning concepts, our idea is that latent temporal signatures found in stations define the common travel patterns of single users. For example, if a station has a peak-outflow at $5pm$, and a user frequently arrives at that station at $5pm$, we infer that the trip purpose of the user likely corresponds to the purpose of the station. For another user with a trip at $4pm$, the weight for the $5pm$-peak temporal signature would not be zero, but a value smaller than the previous user. The small weight value indicates a small probability that this user has a $5pm$-peak travel pattern. This transfer learning approach learns temporal signatures from the station flow matrix decomposition and applies them to decompose the raw user temporal flow matrix $\boldsymbol{U}$. To do this, we implement a multiplicative update rule to project a user flow vector to the latent space defined by the learned temporal signatures as follows:

$$w'^{(i+1)}_{u,k} \leftarrow w'^{(i)}_{u,k} \frac{(\boldsymbol{U}_{u,\cdot}\boldsymbol{H}^T)_{u,k}}{(\boldsymbol{W}'^{(i)}\boldsymbol{H}\boldsymbol{H}^T)_{u,k}}, \tag{4}$$

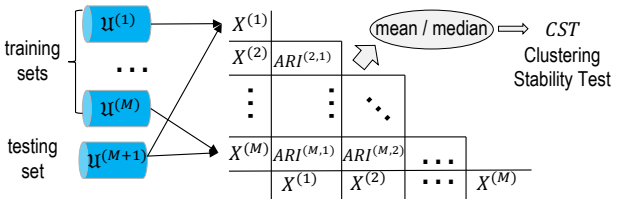


Figure 5: Clustering stability test

where $\boldsymbol{H}$ is the same temporal signature matrix in Equation 1, and $w'_{u,k}$ are weights over shared temporal signatures, that is, an element in all the users' weight matrix $\boldsymbol{W}'$ for user ID u (rows) and temporal signatures k (columns). i is the number of iterations needed until convergence (like Mean Square Error between reconstructed flows and original flows). More details on multiplicative update rules can be found in, e.g., [16, 34]. After finding the weight matrix $\boldsymbol{W}'$ for users, each user is represented by a row vector. We use k-means++ [2] to cluster users based on their temporal signature weights. Since each weight is associated with latent temporal signatures, the semantics of each cluster can be explained by the weights of temporal signatures within the cluster.

4.3.2 ***User Clustering Stability Test:*** To assess the performance of our clustering methods, we introduce a novel domain-specific quantitative evaluation metric called "Clustering Stability Test" (CST) based on the Adjusted Rand Index [14]. The challenge of evaluating our clustering results is that we do not have an authoritative ground truth that defines the "correct clustering". However, we can exploit that the clustering of a set of users should remain the same even if other users are inserted and removed from the database. Specifically, we can leverage that for different training data sets $\{\boldsymbol{U}^{train}_m\}^M_{m=1}$, the same set of tested users $\boldsymbol{U}^{test}$ should be distributed across the same clusters. For example, the cluster of "early bird" commuters in $\boldsymbol{U}^{test}$ should remain the same, regardless of other users $\boldsymbol{U}^{train}_i$ that are additionally considered for clustering. We refer to this as the *stability* of a clustering approach.

Figure 5 summarizes our metric to measure the stability of an algorithm. From left to right we first 1) partition the original user set $\mathcal{U}$ into M non-overlapping training sets $\{\boldsymbol{U}^{train}_m\}^M_{m=1}$ and a non-overlapping test set $\boldsymbol{U}^{test}$. 2) We generate M datasets $\{\boldsymbol{U}_m\}^M_{m=1}$ as the union of each training set with the test set, formally, $\boldsymbol{U}_m = \boldsymbol{U}^{train}_m \cup \boldsymbol{U}^{test}$. The set $\boldsymbol{U}_m$ is clustered and we let X_m denote a set of cluster labels of users in $\boldsymbol{U}^{test}$. The rationale is that we now have M different clusterings of the same group $\boldsymbol{U}^{test}$. 3) We use the Adjusted Rand Index (ARI) [14] to compute the pair-wise clustering similarity score $ARI(X_i, X_j), 1 \le j < i \le M$. This results in a lower triangle matrix of ARI scores. 4) The mean (median) value of all ARI scores is denoted as CST_{mean} (CST_{median}) and is used as our proposed CST score to compare different clustering methods. Additional detials can be found in Appendix E.

5 EXPERIMENTS AND RESULTS

We assess the performance of the proposed S2U Framework by using three real-world datasets and comparing it to two competing methods and one control method. Section 5.1 introduces our experimental setup. Section 5.2 assesses the methods using the

newly-proposed clustering stability procedure *CST*. A qualitative evaluation using spatial-temporal visualizations are discussed in Section 5.3. This includes explainable Metro station generativity and attractivity, and explainable user clustering results that decode urban mobility patterns as part of a Washington D.C. case study.

5.1 Experimental Setup

Real-world datasets: we utilize the following three real-world mobility datasets from Washington D.C.

Metro farecard data was provided to us by the Washington Metropolitan Area Transit Authority (WMATA). The data covers the DC metro area for the week of May-01-2016 to May-07-2016. Each farecard record only contains limited information <Card ID, Entry Station, ArrivalStation, Entry Time, and Arrival Time>. We only consider the entry timestamp, as the trip duration is not part of our model. There are a total of 3.57 million trip records, and about 0.8 million unique card/user IDs. The trip frequency distribution is skewed towards users with a small number of trips per week (< 3 trips). To split the data into a training and testing dataset, we randomly selected without replacements 10 training sets with 50,697 users, each and 1 test dataset with 10,000 users.

Taxi data is collected from different taxi agencies for the open-data initiative of Washington D.C. [3] It includes 89,237 taxi IDs for all taxi services within D.C. during 2016. We treat each taxi ID as a user who serves specific areas. Since taxis do not have stations (at least for drop offs), we use grid cells of size 0.02 ($\approx 2.2km$) by 0.02 degree to create an OD pair temporal flow matrix. We expect tidal traffic patterns for taxi IDs, as a larger demand for transportation during peak hours is expected to lead to an increase in supply offered by taxi companies. However, we note that trip length distribution of different taxi IDs is more uniform than Metro data, since a single driver may now serve many passengers, rather than only his own commute. We randomly split the data into 8 training datasets containing 10,904 taxi IDs each and one test dataset containing 2,000 taxi IDs.

Bike-sharing data was obtained from Capital Bikeshare [4], a service that covers the Greater DC urban area. It includes 3032 unique bike IDs and 401 bike docking stations. Similar to taxi data, there is no specific user ID for each trip, so we chose to use bike IDs as user IDs. In this case, the user clustering pattern does not directly reflect user travel behavior but travelling patterns of different bikes. We use 10 training datasets each of which has 303 bike IDs, and 30 bike IDs for the test dataset. To pre-process the data, raw timestamps are mapped to one 24 hour period for $\mathcal{T}$. Table 1 summarizes the datasets used in the experimentation.

Table 1: Descriptive summaries of experiment datasets

Data	Total users	Total trip	Training sets	Users per train	Users per test
Metro	516,976	845,700	10	50,697	10,000
Taxi	89,237	89,237	8	10,904	2,000
Bike	3,032	51,325	10	303	30

Metric: We use our Clustering Stability Test CST_{mean} and CST_{median} introduced in Section 4.3.2 to evaluate quantitative performance. The higher the clustering stability test score is, the better a method performs. Additionally, since we introduce random splitting to obtain training and test datasets, we need to reduce the chance of an outlying performance from just a random good/bad split. We use Median (MED) (instead of mean) values for a set of CST_{mean} or CST_{median} measures obtained from dozens of runs so as to eliminate the impact of such outliers. The higher a median value is, the better a method performs. We also report the Median Absolute Deviation (MAD) of the same set of runs. MAD tells us how much different random splits could impact the Median value. We expect MAD to be low.

Table 2: Comparisons using user clustering stability test

data	*CST* scores	MED of multiple runs				MAD of multiple runs			
		Naive	NMF	S2U	Control	Naive	NMF	S2U	Control
Metro	CST_{mean} [95% lower] [1]	0.5217 [0.4743]	0.6501 [0.6159]	**0.7019** [**0.6542**]	0.8034 [0.7444]	0.0474	0.0342	0.0477	0.0590
	CST_{median} [95% lower]	0.5504 [0.4981]	0.5815 [0.5482]	**0.6496** [**0.5675**]	0.7347 [0.5947]	0.0523	0.0333	0.0821	0.1400
Taxi	CST_{mean} [95% lower]	0.5417 [0.4951]	0.6605 [0.5801]	**0.8117** [**0.7729**]	1 [1]	0.0466	0.0804	0.0388	0
	CST_{median} [95% lower]	0.4781 [0.4559]	0.6079 [0.584]	**0.8150** [**0.7729**]	1 [1]	0.0222	0.0239	0.0421	0
Bike	CST_{mean} [95% lower]	0.5727 [0.4419]	0.5412 [0.4265]	**0.6347** [**0.5511**]	0.7846 [0.6925]	0.1308	0.1147	0.0836	0.0921
	CST_{median} [95% lower]	0.5697 [0.4404]	0.5525 [0.4356]	**0.6272** [**0.5428**]	0.7816 [0.6532]	0.1293	0.1169	0.0844	0.1284

[1] 95% lower confidence interval is computed by $MED - MAD$.

Competing methods: The following two competing methods and a control experiment are used. Method 1: "Naive" - a naive model using raw trip volumes for each time epoch as clustering features in k-Means++; Method 2: "NMF" - a baseline model using NMF on temporal trip count features proposed in [5] and applying k-Means++ clustering to the reduced weight matrix; Method 3: "Control" - a control experiment, which fully replicates a training set for the clustering stability test. The Control method should output almost perfectly stable clustering, i.e., $CST = 1$ with the objective to show the effectiveness of *CST*. The hyper-parameter tuning of our method is shown in Appendix D.

All experiments were conducted on a Linux workstation with a 10-core processor (i9@3.3GHz) and 64GB of main memory.

5.2 Quantitative Comparison - Clustering Stability Test

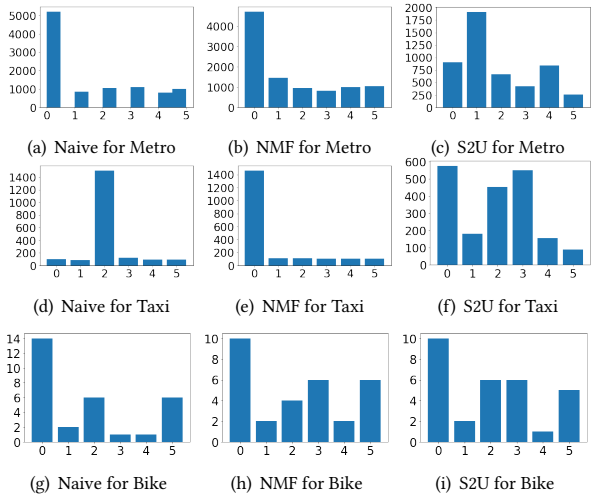
In our experiments, for each method, we conduct a hundred of runs to calculate MED and MAD values, with are the median *CST* value and the deviation based on median for these runs, respectively. Table 2 shows the performance of the different methods. The best MED *CST* scores in each case are indicated in bold and typically stem from our novel S2U method.

Our main finding is that S2U outperforms the other two competing methods for all three datasets in both MED of CST_{mean} and MED of CST_{median}. Even if we subtract the MAD scores from MEDs (e.g., for SFU, CST_{mean} and the Metro data $0.7019 - 0.0477 = 0.6542$), which is the 95% lower bound of the Confidence Interval (CI) [12], S2U still outperforms all other methods, e.g., S2U $CST_{mean} = 0.6542$ for Metro data vs. Naive = 0.5217 and NMF = 0.6501. The Control model shows consistently higher values than the S2U results, instilling confidence in our results.

We also show the distributions of clustering labels for each dataset and each model in Figure 6. An additional proof for a better clustering result is the less skewed distribution of clustering

[3] https://dcgov.app.box.com/v/taxi-trips-2016

[4] https://www.capitalbikeshare.com/system-data



(a) Naive for Metro (b) NMF for Metro (c) S2U for Metro

(d) Naive for Taxi (e) NMF for Taxi (f) S2U for Taxi

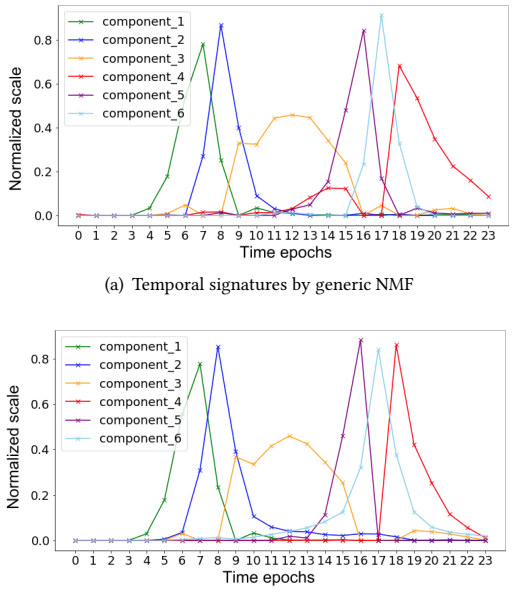
(g) Naive for Bike (h) NMF for Bike (i) S2U for Bike

Figure 6: Histograms of user clustering labels - each bar is a cluster group, total of 6 cluster groups for each method)

labels in Figure 6(c) and Figure 6(f) compared to the Naive and NMF models. This indicates that each cluster captures more meaningful patterns of sparse users. The cluster labels of Metro and Taxi for Naive and NMF models in Figures 6(a), 6(b), 6(d), and 6(e) have all dominating clusters. This is an indicator that these two methods do not capture the real patterns since the raw user temporal flow matrix $\boldsymbol{U}$ and the matrix decomposed by NMF are not informative for sparse users. The clustering labels are more evenly distributed for bikeshare data.

Examining the results in more detail we see that NMF already performs better than Naive for Metro and Taxi data with a 0.13 higher MED CST_{mean} score and 0.03 higher MED CST_{median} score. If we consider MAD values for variances of random splitting, the improvement of CST_{mean} or CST_{median} for NMF is still a significant improvement for the Metro and Taxi datasets when compared to the Naive method. This result tells us that even raw features of Metro users and Taxi drivers contain information that can be used for clustering. Our proposed S2U method provides an even bigger improvement for Taxi data as shown in Table 2. MED CST_{mean} improves significantly by 0.15 and MED CST_{median} improves by an even greater margin of 0.21. Moreover, if we look at Figure 6, S2U found more evenly distributed class labels for Metro and Taxi data, while Naive and NMF are heavily biased towards one group. The former is a good indicator of more useful clusters.

When comparing Taxi and Bike data, we observe that NMF does not perform as well as Naive for Bikeshare data. It is 0.01 to 0.02 lower for MED CST_{mean} and MED CST_{median} scores, respectively. Considering MAD scores and the 95% lower CI, the differences are even smaller, and it is hard to argue that NMF outperforms the Naive method. A possible reason here is that Bike IDs have less of a variance in their pattern than Taxi IDs. Taxi drivers have a unique (spatial) service pattern (commuters) while bikes (we focus on their ids) are more randomly used and distributed throughout the system/area. That is why our S2U framework improves the clustering power by 15 – 21% for Taxi data and only by 7 – 8% for Bikeshare data when compared to the best Naive or NMF result.



(a) Temporal signatures by generic NMF

(b) Temporal signatures by TR-NMF

Figure 7: Temporal signatures by generic Non-negative Matrix Factorization (NMF) and Tidal-Regularized Non-Negative Matrix Factorization (TR-NMF))

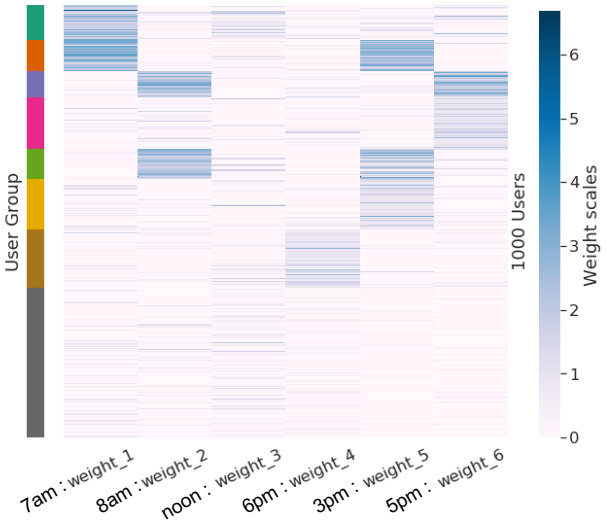


Figure 8: Explainable User Clustering Labels (8 user cluster labels)

5.3 Qualitative Evaluation

This evaluation centers on the qualitative analysis of temporal signatures identified by a generic NMF method and our proposed TR-NMF algorithm. We also provide an intuitive visualization of different user groups' behavior identified by the S2U method. Lastly, we visualize user features transformed to latent space using S2U and raw feature space.

Latent temporal signature improvement: Figure 7 shows the temporal signatures found by TR-NMF and a generic NMF model. This figure plots the normalized total signals for a 24 hour period over time. Figures 7(a) and 7(b) show the temporal signatures found by the generic NMF algorithm and the TR-NMF algorithm, respectively. In both figures, Components 1 & 2 are morning commute

signatures, Components 5 & 6 are afternoon commute signatures, and Components 3 & 4 are non-commute signatures. While the overall trends are quite similar, TR-NMF improves Component 4 (red, one of non-commuting signature) and component 6 (light blue, one of afternoon commute signatures). Component 4 is less pronounced around noon to improve the signal of Component 6 since those temporal features are closer to Component 6's peak feature. This result shows how tidal-regularized loss constrains learning and provides better explainable power for the temporal patterns specifically here for the case of the metro farecard data.

Qualitative evaluation of explainable S2U user clustering: To illustrate the qualitative result and the potential of S2U for explaining user-based urban mobility, Figure 8 visualizes the learned user weights in conjunction with users' cluster labels. Since S2U determines the significance of each temporal signature and respective weight for each user, each temporal signature is strongly associated with specific temporal travel behaviors. We can use this visualization to further analyze user travel patterns.

In Figure 8, each row represents a user's weight vector from $\boldsymbol{W'}$, and each column is a weight for a corresponding temporal signature (we utilize a total of six). In this figure, we randomly select 1000 users. The darker a cell is shaded (dark blue), the larger is its respective weight. To the right of the figure is a color bar that shows weight in relation to shading. On the left side of the figure, eight cluster labels are shown using eight colors. The x axis labels represents the weights that correspond to the respective temporal signatures ("weight_1" corresponds to "component_1") of Figure 7. Overall, we generated eight user clusters, which can help us interpret activities in the DC metro area using farecard data.

For example, users in the dark green cluster (first group from top) have strong early 7am commuting travels, but they do not have pronounced noon or afternoon travel. The brown cluster (second group from top) are users who go to work early, i.e., 7am and also return back home early at around 3pm. The pink cluster (third group from top) contains users who start later at around 8am and return home later at 4pm. The light green clusters (fifth group from top) represents users with early 7am morning and early 3pm afternoon commutes. These three clusters (brown, pink, and light green) capture commuters with different schedules. The purple (fourth group from top) and the yellow clusters (sixth group from top) captures users with less focused morning schedules, but still having an afternoon 3pm or late afternoon 5pm travel pattern, respectively. The dark yellow cluster (seventh group from top) are users who mainly travel in the evening, i.e., around 6pm and capture either night-life users or tourists. The last dark grey cluster is extremely sparse users who have few (1-2) trips per week. The small weight values indicate the low confidence with which they exhibit the various temporal signatures. This cluster also shows the limitations of our approach in that we can categorize users with few trips (3-5 trips per week), but with fewer trips there is simply not enough data to infer the semantics of the user.

Visual user clustering improvement over raw and latent feature spaces: Figure 9 utilizes the t-Distributed Stochastic Neighbor Embedding (t-SNE) [19] to qualitatively show the intrinsic properties of our data. t-SNE is an information-based machine learning visualization technique that can transform a high-dimensional to low-dimensional feature space through non-linear manifold while

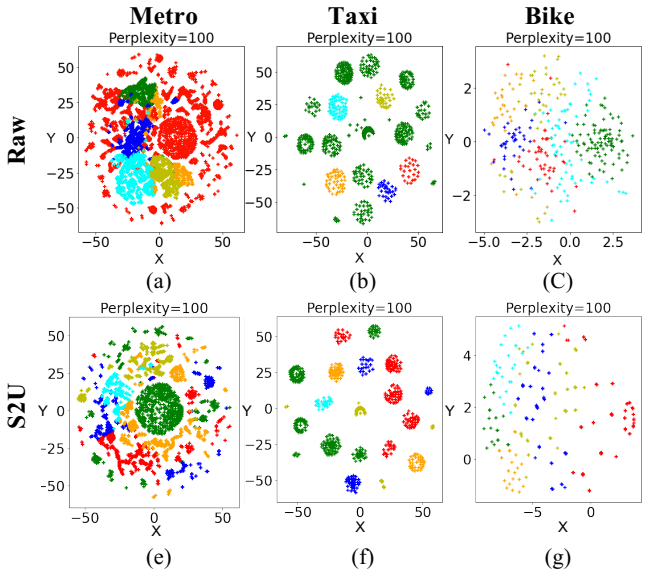


Figure 9: t-SNE visualization in which points are t-SNE transformed users using both raw and S2U-transformed users, and points' colors are based on Naive model and S2U model using 6 clusters.

preserving informative similarity patterns within the data. Figures 9 (a), (b), and (c) (in the top row) show properties of the raw data while the bottom row shows the properties of S2U transformed features. The x and y axis are the data reduced to two features. Each point is a user. Different colors of points are clustering labels identified previously. By comparing raw features and transformed features, we can see that S2U features are more informative and exhibit more distinct patterns. For Metro data, the raw features of users are not very distinctive while the transformed features show clearer clusters. The case is similar for Bikeshare data. For Taxi data, raw features exhibit several distinct clusters. However, S2U made these clusters even more distinguishable. The proposed S2U framework thus boosts the clustering performance even for raw features that already have distinctive clusters. The S2U framework produces clusters that are shown as green, red, and orange points in Figure 9 (d), (e), and (f) (in the lower row). The clusters are more evenly distributed and visually appealing.

Results of generated semantic-based station pattern: Using the tidal-regularized loss, the visualization in Figure 10 shows that our TR-NMF approach provides more explainable station patterns. The sub-figures show station locations (circles) around the area of the White House (the background map). A larger circle indicates stronger attractivity (recovered commuting in-flow for associated temporal signatures). The Metro lines are shown as black lines connecting stations. Figure 10 (a) is based on the 7am commuting signature. Stations around the White House (mostly Federal Government offices) have a higher flow during these early hours. Figure 10 (b) relates to the 8am flow and area such as Dupont Circle (a commercial area) now have a larger flow. Stations around the White House have a comparatively low flow. This example illustrates how using the tidal traffic regularization can support better explainable station patterns and in the future an urban spatial function analysis.

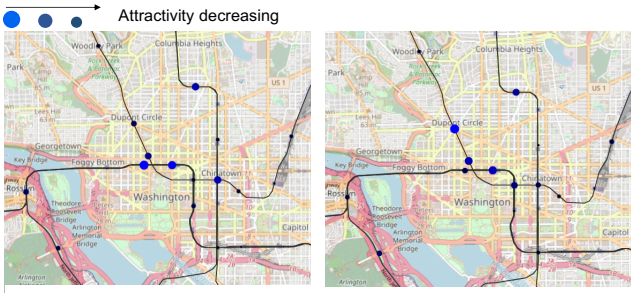


Figure 10: Explainable stations pattern in Washington D.C.

6 CONCLUSIONS AND FUTURE WORK

Inferring semantic information such as trip purpose from basic mobility datasets such as Metro farecard data, Taxi trip data, and Bikeshare data is a challenge given the lack of contextual information with this data. This work proposes a new Station-to-User (S2U) transfer learning framework to achieve a more explainable and stable learning of user clusters from farecard data by transferring users to a latent feature space built with the stations' temporal signatures. As part of the approach, we develop a novel Tidal-Regularized Non-negative Matrix Factorization approach to guide the learning process by including the regular, tidal traffic patterns, e.g., commuters, which dominate urban transportation. To demonstrate the effectiveness of our work, we developed a novel user stability test as an evaluation metric to promote cross-model performance comparison. Lastly, we show that our framework improves the cluster quality in terms CST score by 7% for the challenging Bikeshare data and by 21% for the more cluster-able Taxi trip data. Visualizing the raw datasets and S2U transformed data using t-SNE shows the power of the S2U framework and how it boosts the clusterability of the datasets. Future works includes anomaly detection of users based on travel data over a few weeks/months and urban function analysis using our semantics-based temporal aggregation results.

ACKNOWLEDGEMENTS

This work has been partially supported by the National Science Foundation Grant No. 1637541 and USDOD Grant No. HM02101410004. Liming Zhang has been supported by a presidential graduate research scholarship of George Mason University. We would like to thank the Washington Metropolitan Area Transit Authority (WMATA) for providing the farecard data used in our experiments.

REFERENCES

[1] R. Alvizu, X. Zhao, G. Maier, Y. Xu, and A. Pattavina. Energy efficient dynamic optical routing for mobile metro-core networks under tidal traffic patterns. *Journal of Lightwave Technology*, 35(2):325–333, 2016.

[2] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

[3] R. Becker, R. Cáceres, K. Hanson, S. Isaacman, J. M. Loh, M. Martonosi, J. Rowland, S. Urbanek, A. Varshavsky, and C. Volinsky. Human mobility characterization from cellular network data. *Communications of the ACM*, 56(1):74–82, 2013.

[4] A.-S. Briand, E. Côme, K. Mohamed, and L. Oukhellou. A mixture model clustering approach for temporal passenger pattern characterization in public transport. In *DSAA*, pages 1–10. IEEE, 2015.

[5] L. Carel and P. Alquier. Non-negative matrix factorization as a pre-processing tool for travelers temporal profiles clustering. In *European Symposium on Artificial Neural Networks*, 2017.

[6] Z. Duan, Z. Lei, M. Zhang, H. Li, and D. Yang. Understanding multiple days' metro travel demand at aggregate level. *IET Intelligent Transport Systems*, 2018.

[7] C. Févotte and J. Idier. Algorithms for nonnegative matrix factorization with the β-divergence. *Neural computation*, 23(9):2421–2456, 2011.

[8] B. Furletti, P. Cintia, C. Renso, and L. Spinsanti. Inferring human activities from gps tracks. In *Proceedings of the 2nd ACM SIGKDD international workshop on urban computing*, pages 1–8, 2013.

[9] J. Gan, J. Zhang, and S. Zheng. Where you really are: User trip based city functional zone ascertainment. In *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*, pages 1–8. IEEE, 2018.

[10] Y. Gong, Y. Liu, Y. Lin, J. Yang, Z. Duan, and G. Li. Exploring spatiotemporal characteristics of intra-urban trips using metro smartcard records. In *2012 20th International Conference on Geoinformatics*, pages 1–7. IEEE, 2012.

[11] Y. Han and F. Moutarde. Analysis of large-scale traffic dynamics in an urban transportation network using non-negative tensor factorization. *International Journal of Intelligent Transportation Systems Research*, 14(1):36–49, 2016.

[12] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[13] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5(Nov):1457–1469, 2004.

[14] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[15] T. Lange, V. Roth, M. L. Braun, and J. M. Buhmann. Stability-based validation of clustering solutions. *Neural computation*, 16(6):1299–1323, 2004.

[16] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[17] L. Liu, A. Hou, A. Biderman, C. Ratti, and J. Chen. Understanding individual and collective mobility patterns from smart card records: A case study in shenzhen. In *IEEE Intelligent Transportation Systems*, pages 1–6, 2009.

[18] J. M. Lucas-Cuesta, F. Fernández-Martínez, T. Moreno, and J. Ferreiros. Mutual information and perplexity based clustering of dialogue information for dynamic adaptation of language models. In *Advances in Speech and Language Technologies for Iberian Languages*, pages 148–157. Springer, 2012.

[19] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[20] P. Nitsche, P. Widhalm, S. Breuss, N. Brändle, and P. Maurer. Supporting large-scale travel surveys with smartphones–a practical approach. *Transportation Research Part C: Emerging Technologies*, 43:212–221, 2014.

[21] U. N. D. of Economic and S. Affairs. 2018 revision of world urbanization prospects. https://population.un.org/wup/, 2018. Accessed: 2019-06-09.

[22] R. Oldenburg. *Celebrating the third place: Inspiring stories about the great good places at the heart of our communities*. Da Capo Press, 2001.

[23] M. Poussevin, E. Tonnelier, N. Baskiotis, V. Guigue, and P. Gallinari. Mining ticketing logs for usage characterization with nonnegative matrix factorization. In *Big Data Analytics in the Social and Ubiquitous Context*, pages 147–164. 2015.

[24] A. Rakhlin and A. Caponnetto. Stability of k-means clustering. In *Advances in neural information processing systems*, pages 1121–1128, 2007.

[25] T. Reed. Inrix global traffic scorecard, 2019.

[26] M. A. Taylor. Network modelling of the traffic, environmental and energy effects of lower urban speed limits. *Road & Transport Research*, 9(4):48, 2000.

[27] E. Tonnelier, N. Baskiotis, V. Guigue, and P. Gallinari. Smart card in public transportation: Designing a analysis system at the human scale. In *IEEE Intelligent Transportation Systems*, pages 1336–1341, 2016.

[28] S. Troia, G. Sheng, R. Alvizu, G. A. Maier, and A. Pattavina. Identification of tidal-traffic patterns in metro-area mobile networks via matrix factorization based model. In *IEEE PerCom Workshops*, pages 297–301, 2017.

[29] J. Wang, X. Kong, F. Xia, and L. Sun. Urban human mobility: Data-driven modeling and prediction. *ACM SIGKDD Explorations Newsletter*, 21(1):1–19, 2019.

[30] J. Wang, J. Wu, Z. Wang, F. Gao, and Z. Xiong. Understanding urban dynamics via context-aware tensor factorization with neighboring regularization. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[31] P. Wang, Y. Fu, G. Liu, W. Hu, and C. Aggarwal. Human mobility synchronization and trip purpose detection with mixture of hawkes processes. In *ACM KDD*, pages 495–503, 2017.

[32] S. Wang, Y. Xu, and S. Gao. Revealing functional regions via joint matrix factorization based model. In *2016 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, pages 205–209. IEEE, 2016.

[33] Y. Wang, Y. Zheng, and Y. Xue. Travel time estimation of a path using sparse trajectories. In *ACM KDD*, pages 25–34, 2014.

[34] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *ACM SIGIR*, pages 267–273. ACM, 2003.

[35] C. Yang, F. Yan, and X. Xu. Daily metro origin-destination pattern recognition using dimensionality reduction and clustering methods. In *IEEE ITSC*, pages 548–553. IEEE, 2017.

[36] J. Yuan, Y. Zheng, and X. Xie. Discovering regions of different functions in a city using human mobility and pois. In *ACM KDD*, pages 186–194, 2012.

[37] K. Zhang, Q. Jin, K. Pelechrinis, and T. Lappas. On the importance of temporal dynamics in modeling urban activity. In *ACM SIGKDD International Workshop on Urban Computing*, pages 1–8, 2013.

A LEMMA FOR SWAPPING LINES FOR NMF

Lemma A.1. *Let $\boldsymbol{W} \in R^{m\times K}$, $\boldsymbol{H} \in R^{K\times n}$. Further, let $x, y \leq K$, let $\boldsymbol{W'}$ be obtained by swapping lines x and y in $\boldsymbol{W}$, and let $\boldsymbol{H'}$ be obtained by swapping corresponding columns x and y, then*

$$\boldsymbol{WH} = \boldsymbol{W'H'}$$

Proof. Let $\boldsymbol{V} = \boldsymbol{WH}$, and let $\boldsymbol{V'} = \boldsymbol{W'H'}$. For any cell v_{ij} in $\boldsymbol{V}$ is derived by matrix multiplication as

$$v_{ij} = \sum_{k=1}^{K} w_{ik}h_{ki} = \sum_{k=1,k\neq x,y}^{K} w_{ik}h_{ki} + w_{xk}h_{kx} + w_{yk}h_{ky}$$

Equivalently, we obtain

$$v'_{ij} = \sum_{k=1}^{K} w_{ik}h_{ki} = \sum_{k=1,k\neq x,y}^{K} w_{ik}h_{ki} + w_{yk}h_{ky} + w_{xk}h_{kx}$$

Since $w_{xk}h_{kx} + w_{yk}h_{ky} = w_{yk}h_{ky} + w_{xk}h_{kx}$ by commutativity of multiplication, we get $v_{ij} = v'_{ij}$ for any $i, j \leq k$. Thus $\boldsymbol{V} = \boldsymbol{V'}$. □

This swapping of lines using Lemma A.1 allows us to assume, without loss of generality, that columns of $\boldsymbol{W}$ and lines of $\boldsymbol{H}$ are grouped into morning features first and afternoon features last.

B COMPLEXITY ANALYSIS

The complexity of our algorithm only adds constant factor to generic NMF analysis [7] $O(kMN)$ in theory, where k is number of latent components, M is number of rows of $\boldsymbol{W}$ matrix, N is number of columns of V matrix. As in our case, M is the number of station pairs, which equals $|\mathcal{OD}|$ in $O(S^2)$ and N is the number of time slots $|\mathcal{T}|$. The run-time complexity of our proposed tidel-regularized NMF only adds a constant factor c to generic NMF and lies in $O(ck|\mathcal{OD}||\mathcal{T}|) = O(k|\mathcal{OD}||\mathcal{T}|)$. This constant factor of additional complexity over NMF model training comes from updating gradients of the proposed tidal-regularized loss without affecting the total gradient updating iterations. Kmean++ [2] has the same run-time complexity as KMeans $O(K * N * D)$, where K is the number of clusters, N is the number of users $|\mathcal{U}|$, and D is the dimensionality of feature vectors which is K in our case. To total, the run-time of our algorithm lies in $O(k|\mathcal{OD}||\mathcal{T}| + K|\mathcal{U}|D)$.

C EARLY STOPPING

To reduce running time, a common approach is to do early stoping which terminates the gradient updating loop when the difference of weight matrix $\boldsymbol{W}$ in t^{th} step and in $t-1^{th}$ weight is small than an very small threshold ϵ. $||\boldsymbol{W}^t - \boldsymbol{W}^{t-1}||_2 < \epsilon$. This does not change our theoretical complexity analysis.

D HYPER-PARAMETER TUNING

There are a few hyper-parameter we need to choose. α, η in generic NMF loss, and γ, ρ for TR-loss. We use $\alpha = 1.0, \eta = 0.9, \gamma = 0.1, \rho = 0.1$ in our presented results. We run 100 loop for gradient updating. Other parameters includes number of latent components, number of time slots, number of morning and afternoon subsets .etc, which we leave to the future work. Notice that our approaches are two-fold: transfer learning and TR-loss. Others could use either transfer learning without TR-loss or the other way around.

E USER CLUSTERING STABILITY TEST

Different metrics such as potential (sum of squared distances of samples to their closest cluster center) [12], log-likelihood score [12], perplexity score (information measure of generative probabilistic models) [18], AIC [12], and BIC [12], are used to assess clustering quality based on model assumption or information theory. However, they are not able to assess the stability of a clustering. Various works exist to test the stability of clustering, e.g., [15, 24]. Our proposed metric is based on the Adjusted Rand Index (ARI) [14]. Generic ARI score is computed in this way: for a dataset, like users $\mathcal{U}$, one clustering result assigns a set of group labels to each user with $X = \{x_1, x_2, \ldots, x_u\}$, while another clustering result assigns a set of labels $Y = \{y_1, y_2, \ldots, y_u\}$. An ARI score $ARI_{x,y}$ is computed based on these two label sets with random permutation of cluster label orders (cf. [14]). $ARI_{x,y}$ is a value with a range of $[-1, +1]$, where 0 indicates complete random labeling, +1 stands for a perfect match, and -1 indicates complete reversed labeling.

The proposed metric is named "clustering stability test", which first partitions the original user set $\mathcal{U}$ into non-overlapping M training sets $\{\mathcal{U}^{(m)}\}_{m=1}^{M}$ and a non-overlapping testing set $\mathcal{U}^{(M+1)}$. Then, an end-to-end procedure is applied to a mixed set that joins a training set with the testing set, $\mathcal{U}'_m = \mathcal{U}^{(m)} + \mathcal{U}^{(M+1)}, \forall m \in 1, \ldots, M$. A label set $X^{(m)}$ is created for $\mathcal{U}'_m$ using S2U. For each pair of $X^{(m_i)}$ and $X^{(m_j)}$, we calculate the score $ARI^{(m_i,m_j)}$. The Mean value of all the paired ARI scores, denoted as CST_{mean}, is used as the "clustering stability score" to compare different clustering methods. Another choice is to use the Median value of all the scores $ARI^{(m_i,m_j)}$ denoted as CST_{median}. This score has the range of $[-1, +1]$, with +1 indicating a perfectly stable clustering method.