PROGRESS REPORT ON STUDY OF NYC TRAFFIC

VAIBHAV KARVE
DERREK YAGER

1. Restricting to a block

We restrict our attention to the block between W. 44th to 45th St. and 8th to 9th Ave. This block was selected at random. On extracting the data from the data file for 2011, we see that these streets are all one-way access only.

1.1. **Naming convention.**

1.1.1. *Intersections.*

- A = W. 44th St. and 8th Ave. (Node_id = 42435671)
- B = W. 44th St. and 9th Ave. (Node_id = 42443561)
- C = W. 45th St. and 8th Ave. (Node_id = 42432700)
- D = W. 45th St. and 9th Ave. (Node_id = 42432703)

1.1.2. *Roads.*

- BA (Link_id = 128255)
- AC (Link_id = 169017)
- CD (Link_id = 182993)
- DB (Link_id = 181188)

We restrict our attention to only these links, one at a time. For each link, from the database, we extract 2 separate arrays: one giving the average travel time in seconds, for every hour of the year; and the other giving the number of trips on that used that link, for every hour of the year.

1.2. **Periodicity analysis.**

1.2.1. *Stratifying the data.* Intuitively, one may expect that traffic patterns repeat themselves every 7 days (weekly) or maybe every 30 days (monthly). Or perhaps, there is no such periodicity. Whatever be the case, the periodicity should not be imposed, but rather should be inferred from the data itself. To do so, we we assume a particular period in days, call it `period` and divide the entire data into 24 × `period` number of bins. This converts our data from flat lists of trips and traveltimes to something that may be viewed as stratified data.

Stratified data now looks like:

$$\begin{array}{cc}
 & \begin{array}{cccc}\text{Cycle 1} & \text{Cycle 2} & \cdots & \text{Cycle } \left(\left\lfloor\frac{365}{\texttt{period}}\right\rfloor+1\right)\end{array}\\
\begin{array}{c}\text{Stratum 1}\\ \text{Stratum 2}\\ \vdots\\ \text{Stratum 24*}\texttt{period}\text{-1}\\ \text{Stratum 24*}\texttt{period}\end{array} & \begin{pmatrix} 5 & 2 & \cdots & 10\\ ? & 7 & \cdots & 5\\ \vdots & \vdots & & \vdots\\ 33 & ? & \cdots & ?\\ ? & 32 & \cdots & ?\end{pmatrix}
\end{array}$$

Date: July 10, 2016.

Here, number of rows $= 24 \times \texttt{period}$. We let period range from 2 to 38 for our periodicity analysis. As an illustration, we show below the special case when $\texttt{period} = 7$ i.e., cycles are weeks.

$$\begin{array}{l|cccc|}
 & \text{Week 1} & \text{Week 2} & \cdots & \text{Week 53} \\
\text{Sat 0000-0100 hrs} & 5 & 2 & \cdots & 10 \\
\text{Sat 0100-0200 hrs} & ? & 7 & \cdots & 5 \\
\vdots & \vdots & \vdots & & \vdots \\
\text{Fri 2200-0300 hrs} & 33 & ? & \cdots & ? \\
\text{Fri 2300-0000 hrs} & ? & 32 & \cdots & ?
\end{array}$$

Note that the trailing values in the last column will need to be set to (?) because a year does not contain 53 full weeks and there will be data points "missing" in the last week.

1.2.2. *Missing values.* The question marks (?) in the above matrix correspond to those hours for which we have no data on our link. This ofcourse does not mean that there is no traffic on that link at that time, just that we don't know what it it. These are to be treated as missing values in our data and need to be somehow inferred.

The links BA, AC, CD and DB have 20%, 0%, 0% and 1% of their values missing, respectively.

The most natural first approximation for these missing values is the mean value of the corresponding stratum to which each missing value belongs.

The Inferred Stratified data looks like:

$$\begin{array}{c|cccc|}
 & \text{Cycle 1} & \text{Cycle 2} & \cdots & \text{Cycle } \left(\lfloor\frac{365}{\texttt{period}}\rfloor + 1\right) \\
\text{Stratum 1} & 5 & 2 & \cdots & 10 \\
\text{Stratum 2} & Mean_1 & 7 & \cdots & 5 \\
\vdots & \vdots & \vdots & & \vdots \\
\text{Stratum 24*}\texttt{period}\text{-1} & 33 & Mean_2 & \cdots & Mean_n \\
\text{Stratum 24*}\texttt{period} & Mean_1 & 32 & \cdots & Mean_n
\end{array}$$

where $n = \left(\lfloor\frac{365}{\texttt{period}}\rfloor + 1\right)$. Here, we calculated the mean for each stratum by ignoring the missing values.

1.2.3. *Stratified variances.* To establish the optimal choice of `period` for the data, we calculate the variance of the inferred stratified data we obtained in the previous subsection.

$$\text{Variance of stratified data} \approx \frac{1}{n}\sum_{n} \text{Variance of each stratum}$$

where $n = 24 \times \texttt{period}$ is the total number of strata.

If there truly does exist a periodicity in the data, for the correct `period`, the values within each stratum will lie close to each other and hence, the variance will be minimized. Hence, we look for dips in the variance.

1.2.4. *Conclusion of periodicity analysis.* We calculate the variance for each value of `period` from 2 to 38. The graph of the variances for link BA for the No. of Trips during 2011 is given below:

The other three links show similar dips in variance at `period` values which are multiples of 7, in both data – Travel times as well as No. of Trips. **Conclusion:** NYC traffic data has a 7-day periodicity.

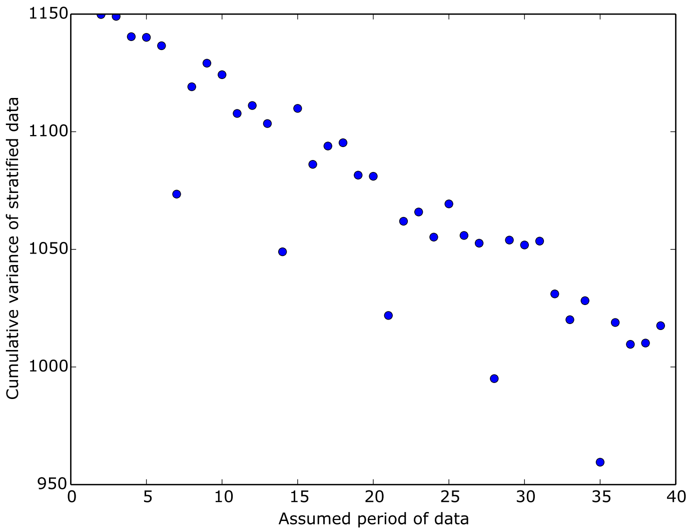


FIGURE 1. The dips in variance correspond to `period` $= 7, 14, 21, 35$.

1.3. **Periodicity analysis via autocorrelation.** An easier, and more standard way to establish periodicity would be to calculate the autocorrelation of our data for four links assuming different lags (`period` values) of 1 to 39 days.

For a discrete set of observations $\{X_1, X_2, \ldots, X_n\}$, an estimate of the autocorrelation can be obtained as,

$$\widehat{\text{Autocorrelation}}(k) = \frac{1}{(n-k)\sigma^2} \sum_{t=1}^{n-k} (X_t - \mu)(X_{t+k} - \mu)$$

where $k = 24 \times$ `period` is the lag in hours for which we are testing autocorrelation in our data, and μ and σ^2 are the mean and variance of the data respectively.

1.3.1. *Missing values.* We calculate the mean (μ) of the data, ignoring the missing values. We then replace the missing values with μ. We calculate the variance (σ^2) of this *corrected* data. We then proceed to calculate the autocorrelation for different values of `period` using the formula mentioned in the previos subsection.

1.3.2. *Conclusion of autocorrelation study.* We plot the autocorrelations for Trips data and Traveltimes seperately, for all four road links simultaneously. The autocorrelation is high for `period` values that are a multiple of 7. This confirms what was seen by the statifying the data – NYC traffic patterns repeat every seven days.

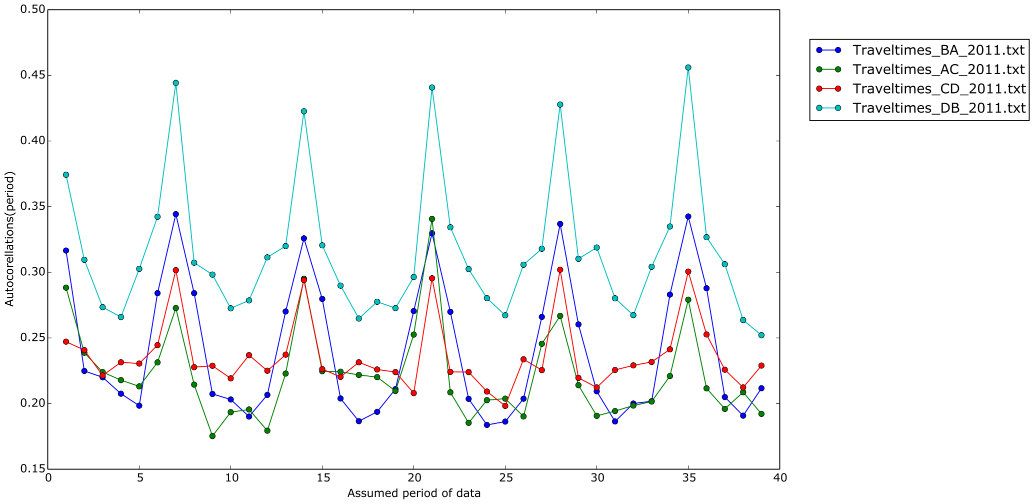


FIGURE 2. Autocorrelations for Travel-times data of all four road links. The peaks in autocorrelation correspond to `period` $= 7, 14, 21, 35$.

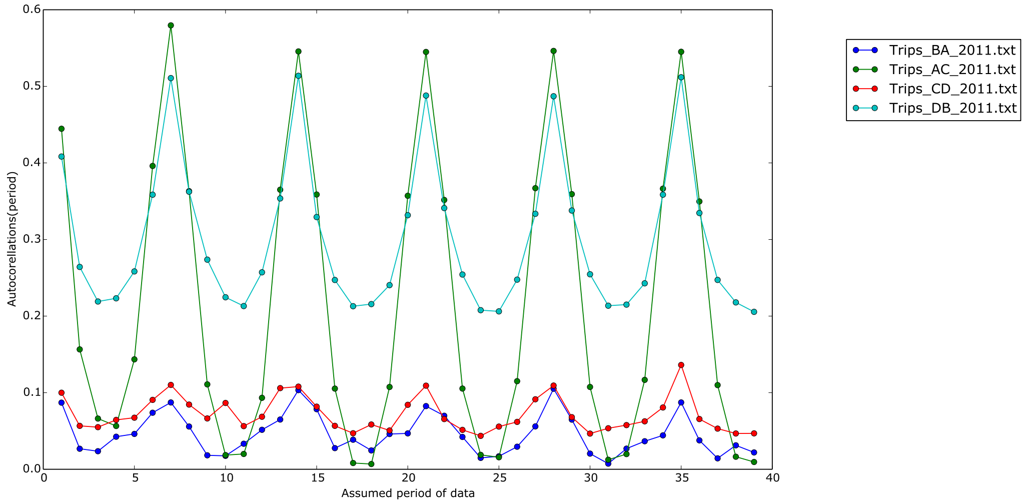
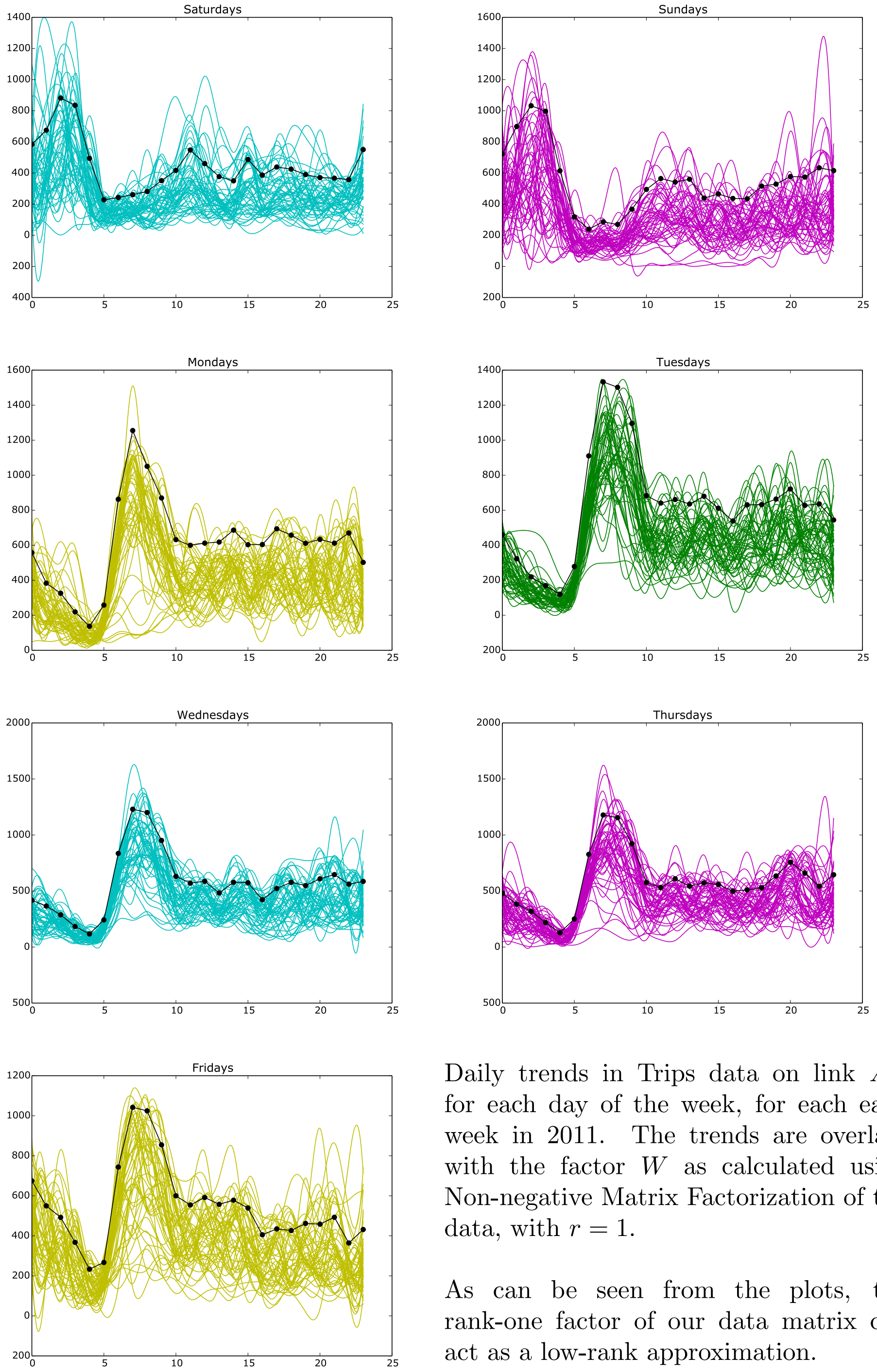


FIGURE 3. Autocorrelations for Trips data of all four road links. The peaks in autocorrelation correspond to `period` $= 7, 14, 21, 35$.

1.4. **Visualizing the data.** Keeping in view the conclusions drawn from the autocorrelation analysis, henceforth, we try to analyze each dat of the week seperately. We try to visualize the data by plotting different parts of the data file, in the hopes that it will point towards directions in which we can carry out further study.

1.4.1. *Daily pattern.* Below are shown the graphs for each day of the week. These plots are also overlaid with the $r = 1$ results from Non-negative Matrix factorization. NMF is discussed in greater detail in the next section.



Daily trends in Trips data on link AC for each day of the week, for each each week in 2011. The trends are overlaid with the factor W as calculated using Non-negative Matrix Factorization of the data, with $r = 1$.

As can be seen from the plots, the rank-one factor of our data matrix can act as a low-rank approximation.

Unresolved issue: Why is the W vector slightly higher than the “mean” daily trend?

1.4.2. *Observations from visual examination of daily trends.* From visual examination of the graphs showing daily trends of traffic, we can make a few observations:

(1) After superimposing the plots for each day of the week, one can clearly see a daily pattern. For example, on an average Monday:
   - Midnight to 1am: the traffic drops to a minimum value at 1am, settling down for the night.
   - 1am to 5am: very thin traffic.
   - 5am to 10am: very high traffic. Traffic peaks at 7am.
   - 10am to midnight: traffic plateaus at a value less than rush hour. This region has very high variance across all Mondays i.e. the exact traffic in this time-period varies a lot from Monday-to-Monday.

(2) The schedule outlined above holds for all the weekdays, i.e. Monday-Friday behave the same.

(3) The patterns for both Saturday and Sunday are very similar to each other, but different from the weekday schedules:
   - Midnight to 3am: high traffic, with high variance across all weekends. Traffic peaks at 2am.
   - 3am to 5am: traffic drops drastically and settles at a minimum value at 5am.
   - 5am to 9am: very thin traffic.
   - 9am to midnight: traffic plateaus at a value less than rush hour. This region has very high variance across all weekends i.e. the exact traffic in this time-period varies a lot from weekend-to-weekend.

(4) On calculating the derivative matrix for each day of the week, one can see that those plots show similar weekday vs. weekend patters. (Graphically though, the derivative matrix's plot is more difficult to interpret compared to the actual transport matrix's plot.)

(5) Regions with high variance: As pointed out in 1. and 3., certain periods have a high variance even after taking into consideration the weekly periodicity in data. These might point to other factors that need to be considered eg: seasonal variations in traffic.

(6) All the above analysis was done with data from a single road link. However, other links show similar (but not completely identical) patterns.

(7) Item 6. points to what might be the most important and counter-intuitive aspect of all: these patterns in traffic are present not just globally but even locally. One might expect that even though we may see some small patterns when studying a single link, more complex phenomena like rush-hour, weekend vs. weekday patterns etc. would not be visible at the street-level and would show up only once we start considering larger networks of roads. However, items 1. to 4. go towards showing that these phenomena percolate down to the street level!

1.5. **Non-negative Matrix Factorization.**

1.5.1. *The theory.* Suppose we have a large matrix D of datapoints which we wish to understand. It is easier to handle this data, and make sense of it if the dimension can somehow be reduced. One of several matrix factorization methods can be used to achieve this dimensionality reduction.

Given a $n \times m$-matrix D, we wish to find matrices W and H of sizes $n \times r$ and $r \times m$ respectively satisfying

$$D \approx WH$$

Here, the choice of r is up to us. Matrix factorization is useful only when we choose $r < \min\{n, m\}$. There exist several algorithms, starting from an initial guess of W

and H, iteratively update these matrices, to guarantee convergence to D in a finite number of steps.

Moreover, since all our datapoints (i.e. matrix entries) either consist of road-links, average speeds, or average travel times, in order to make sense of the matrix factors, it is desirable to constrain the elements of W and H to always be non-negative. This can be guaranteed by the choice of iterative updatation rules. We choose the algorithm called Non-negative Matrix Factorization (NMF) for this very reason.

1.5.2. *The algorithm.*

(1) Choose a value for $r < \min\{n, m\}$, where $n \times m$ is the dimension of D.
(2) Initialize W and H as random matrices, with sizes $n \times r$ and $r \times m$ respectively.
(3) Define error as

$$\begin{aligned}\text{Error} &= ||D - WH||^2 \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m}(D_{ij} - (WH)_{ij})^2\end{aligned}$$

(4) In order to find a local minima of Error, iteratively update W and H using the following rules:

$$W_{ij} \leftarrow W_{ij}\frac{(DH^{\mathsf{T}})_{ij}}{(WHH^{\mathsf{T}})_{ij}}$$

and

$$H_{ij} \leftarrow H_{ij}\frac{(W^{\mathsf{T}}D)_{ij}}{(W^{\mathsf{T}}WH)_{ij}}$$

(5) Stop when Error gets smaller than a chosen tolerance level.

1.5.3. *Implementing the algorithm.* For the sake of explaining how we implement NMF in our dataset, we restrict our attention to only the Trips data for Link AC in the year 2011. This can be written out as a single vector of size 8760, starting at the trips data for January 1, 12am-1am and ending December 31, 11pm -12am.

In view of the autocorrelation results showing us a weekly-periodicity in traffic data, we can stack the Trips data such that each week is respresented in a seperate column. The resulting matrix, which we call D looks like this:

$$D = \begin{array}{r|cccc|} & \text{Week 1} & \text{Week 2} & \cdots & \text{Week 53} \\ \text{Sat 0000-0100 hrs} & * & * & \cdots & * \\ \text{Sat 0100-0200 hrs} & * & * & \cdots & * \\ \vdots & \vdots & \vdots & & \vdots \\ \text{Fri 2200-0300 hrs} & * & * & \cdots & 0 \\ \text{Fri 2300-0000 hrs} & * & * & \cdots & 0 \end{array}$$

Note that the trailing values in the last column will need to be set to zero because a year does not contain 53 full weeks and there will be data points “missing” in the last week.

1.5.4. *Conclusions from NMF.* Implementing the NMF algorithm for $r = 1$ yeilds a vector W which accounts for the data to a great extent, as seen in the in the graphs showing trends.

1.6. Unresolved issues and things that need further investigation.

(1) Regions with high variance: As pointed out in 1. and 3., certain periods have a high variance even after taking into consideration the weekly periodicity in data. These might point to other factors that need to be considered eg: seasonal variations in traffic.

(2) In the graphs, the W vector is slightly higher than the "mean" daily trend, i.e. W seems to be enveloping the traffic trend rather than approximating it. The reason for this is unclear.

(3) While we have made use of the $r = 1$ NMF, it is not clear how the results from higher-rank factorizations is to be interpreted.

(4) The NMF analysis outlined above was done only for link AC because AC has almost no missing data. We need to figure out a way to fill in the missing data before applying NMF to other links. Or perhaps, we need to modify the NMF algorithm to infer the missing datapoints from the factors themselves. (Weighted matrix entried need to be explored as a way of doing this.)

2. 3-Point Plan

All the data for an entire year can be organized into a matrix of size 8760×260855. The 8760 comes from 24 × 365 while 260855 represents the number of links on the NYC grid. But most links on the grid have a lot of data missing on them. We therefore divide the job of estimating traffic on them into three different phases.

2.1. Phase 1: Compression.

We first deal with links that have less than 30-days worth of data missing. This means, we restirct our attention to only 2302 links.

Using NMF we factor D, which is a 8760×2302 matrix into W and H of sizes $8760 \times \text{rank}$ and $\text{rank} \times 2302$, rank is a variable that we need to pick.

NMF is aimed at minimizing the error $||D - WH||^2 = \sum_{i,j}(D - WH)^2_{ij}$. As seen in the following figure, relative error drops with increase in rank and we therefore should aim for a higher rank in order to get a closer approximation.

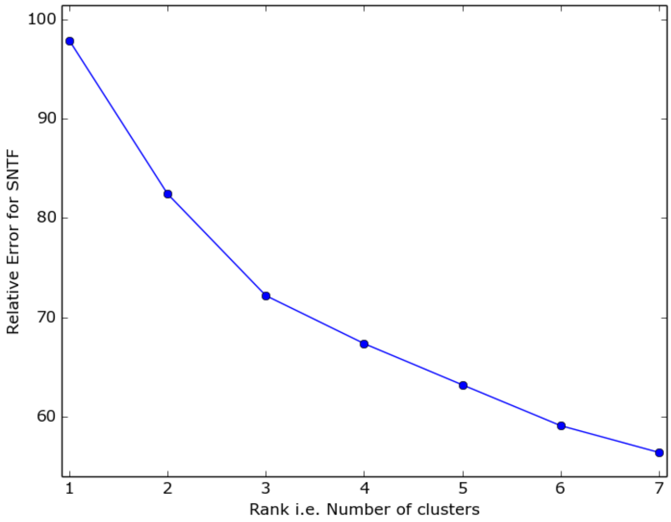


FIGURE 5. SNTF relative error vs. Rank

However, there is a price to pay for picking a higher rank in that higher rank makes for poorer compression. Each unit increment in rank would mean we need to store 8760 additional numbers in W and 260855 additional numbers in H.

As will be discussed shortly, we can save on the H-entries by running Sparse-NMF instead of NMF. However, the fact remains that each increment in entails atleast 8760 additional entries. So what should be the optimal choice? How do we make sure that a higher choice of rank is not simply fitting some white noise already present in D? Some of these questions can be answered by Sparse-NMF, while also achieving better compression and pattern-recognition.

2.1.1. *Sparse Non-negative Matrix Factorization.* Sparse-NMF (hereafter SNMF) minimizes the following error:

$$\text{Error} = ||D - WH||_2^2 + \beta \sum_i ||H[:, i]||_1^2 + \eta ||W||_2^2$$

where $H[:, i]$ represents the i^{th} column of H and β, η are positive constants that we need to fix.

To achieve this, we just use different multiplicative update rule. SNMF ensures sparsity in H, i.e. it adjusts the columns of H to look more like $(1, 0, \dots, 0)$, $(0, 1, 0, \dots, 0)$ and so on.

SNMF clusters every column of D into several clusters. The columns of W record the trend of each cluster centroid while the columns of H record the projection of the trend on a certain link onto each of the clusters. For example, an H-column of $(0, 3, 0, \dots, 0)$ would mean that traffic on the given link matches cluster #2 and that the approximate trend can be obtained by scaling the second column of W by a factor of 3.

We define a new metric to measure the sparsity of H:

$$\text{Sparsity} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\text{rank}} \left\| \frac{H[:, i] - \max H[:, i]}{\max H[:, i]} \right\|_2^2$$

A completely sparse H would have Sparsity = 1. In practice though, a sparsity value between 0.6 to 1 can be considered to be sufficient.

Note that Sparsity of H will be affected by choice of β and η.

2.1.2. *Choosing rank.* We need to choose the smallest rank that keeps the error below some given threshhold. But since it is difficult to decide what this threshhold should be, we choose an alternate strategy. We apply SNMF on D for various values of rank. Seeing as SNMF clusters links based on their traffic pattern, we keep increasing rank till the point where we start getting empty clusters. Beyond this point, adding more rank will be superfluous because it will not change the clustering pattern even though it might keep reducing the error via noise-fitting.

We found that for Trips data for the year 2011, this limit was reached at rank = 8. Fixing rank at 8, we can try to choose β and η to get the sparsity of H as close to 1 as possible.

2.2. **Phase 2: Estimation.** From Phase 1, we obtain the 8 typical traffic trends. The underlying assumption is that even though these 8 trends were picked out after analyzing the yearly trends of 2303 of the busiest links in New York, all other 258552 remaining links follow one of these eight trends (up to a scale factor). We now need to assign the correct cluster to each new link without recalculating the clustering pattern each time. We do this simply by minimizing the Euclidean distance of each trend from the trend of every cluster cetroid.

For each new link, given a trend vector, T of length 8760 with possibly missing data having been filled with the mean of the available values, we wish to find $c > 0$ and $i \in \{1, 2, \dots, 8\}$ such the solve the following minimization problem:

$$\min_{i \in \{1,2,\dots,8\}} \min_{c>0} ||T - cW_i||_2^2$$

where W_i denotes the i^{th}-column of W.

To find c, we solve

$$\begin{aligned}
0 &= \frac{\partial}{\partial c} ||T - cW_i||_2^2 \\
&= \frac{\partial}{\partial c} \big\langle (T - cW_i)^{\mathsf{T}}; T - cW_i \big\rangle \\
&= \frac{\partial}{\partial c} \big\langle T^{\mathsf{T}} - cW_i^{\mathsf{T}}; T - cW_i \big\rangle \\
&= \frac{\partial}{\partial c} \big(||T||_2^2 + c^2 ||W_i||_2^2 - 2c\langle T^{\mathsf{T}}; W_i \rangle \big) \\
&= 2c||W_i||_2^2 - 2\langle T^{\mathsf{T}}; W_i \rangle
\end{aligned}$$

Therefore,

$$c = \frac{\langle T^{\mathsf{T}}; W_i \rangle}{||W_i||_2^2}$$

Once we have c, we just pick the $i \in \{1, 2, \dots, 8\}$ that minimizes the Euclidean distance. In this manner, we can assign clusters to each new link one-by-one.

2.3. **Phase 3: Folding.** Phase 2 worked around the problem of having missing values in the data by simply filling in with the mean data-entry for that given link. However, there are links which have very little or almost no data on them. In such cases, filling in with a mean presents two difficulties: firstly, it results in a near-constant trend and hence makes for a bad fit in Phase 2. Secondly, the clustering become very sensitive to our guess of the ideal filler value.

To get around this problem, we exploit patterns/redundancies that we know exist in our data. For example, we have four-years worth of data ($2010 - -2013$). We can use this redundancy to increase the number of data entries on our link. We could also use the 7-day periodicity in data to replicate available data-points before following through with Phase 2.