Analysis of Migratory Relationship between Tier 1 Cities and States of India using Twitter Data

Arsh Gautam, Ankit Kumar Singh

June 13, 2021

1 Introduction

In India, the development is pyramidal. On the top of this pyramid lie the top industrial cities. These cities attract domestic as well as international companies to set up their businesses. A large pool of jobs are concentrated in these cities. There is a significant difference between the income level, lifestyle, infrastructure etc. between these cities and the Rest of India.

Due to these reasons, these cities have a large inflow of domestic migrants from across the country. The migrants constitute from laborers, factory workers, domestic workers, etc. to educated salaried people. Along with this, In the past decade, there has been a rapid advancement of technology as well as access to smartphones and social media to a large number of common people in India.

A result of a study shown below shows the approximate number (projected for future) of smartphone users in India :

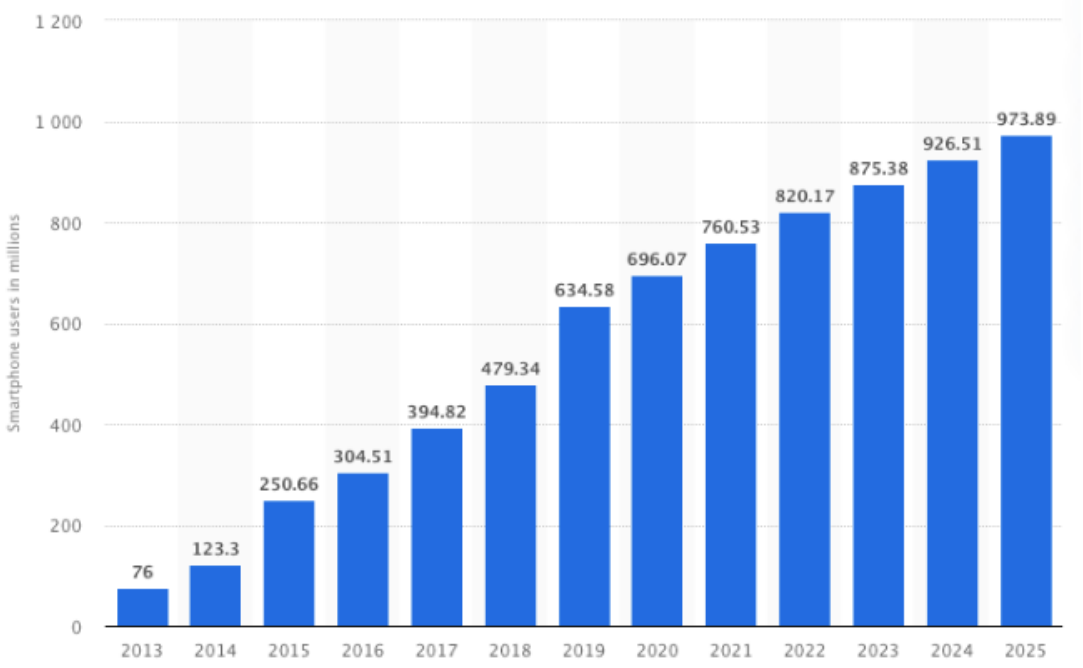

Figure 1: Number of smart phone users in India [10]

Thus, it seems a good idea to analyse the social media statistics to get an idea about the migration pattern in India. In this study, we aim to use social media, twitter in particular, to get a sense of the strength of migration between various cities and states selected carefully on

the basis of inbound migration and out migration data. We then try to compare our findings with those obtained from Census data.

2 Related Work

Migration patterns in India have been extensively studied. A study (Ansary, 2018 [3]), discusses the changing patterns in Indian migration space analysing the Census 2011 and 2001 data using statistical techniques and drawing a comparison between the two. Another study (Taralekar, et al., 2012 [7]), also uses Census data to assess the pattern of migration in India. They use the Census 2001 and 1991 data. The country is divided into multiple zones and then zone-wise classification of internal as well as international migrants is studied. They correlate the migration to socio-economic factors like GNI per capita, literacy rate, population density and urbanization in their study.

These studies on Indian Migration prominently rely on the Census data. We aim to approach this problem from another direction. We use Twitter data and apply various machine learning models to the data to obtain our results. We use the technique of vectorization of tweets using different methods like DocTag2Vec and TF-IDF, and use these vectors to build our models. These vectors capture the information present in natural language in a vector space. We then further build our hypothesis to analyse these vectors building multiple models over them. Moreover the study of migration between cities and states is another novelty in our work.

In some international studies, Twitter has been used to study various forms of migration. In such a study (S Urchs, et al., 2019 [4]) presents a dataset of 3275 tweets, annotated regarding their relevance of con-taining quantitative movement information of refugees/migrantsinto Hungary, Austria and Germany. Another exciting study[9] aims to analyse international migration using Twitter data clubbed with Facial Recognition techniques. A more closely related study to ours (C Armstrong, 2021 [2]) comments on the challenges when analysing migration on the basis of geolocated twitter data. In previous studies migration has been tried to be studies assigning different users as migrants or not on the basis of their tweet activities and Armstrong et al. state "Upon close inspection very few of the accounts that are classified as migrants appear to be migrants in any conventional sense or international students". We use a novel method of predicting migration strength on basis of tweet texts rather than assigning users any labels.

3 Dataset

There are a total of 18 cities as well as 18 states that we have shortlisted for our study.

Cities: New Delhi, Hyderabad, Kolkata, Lucknow, Bangalore, Mumbai, Chandigarh, Nagpur, Bhopal, Jaipur, Chennai, Gandhinagar, Gurgaon, Tiruchirappalli, Visakhapatnam, Cuttack, Warangal, Pune.

States: Rajasthan, Odisha, Tripura, Meghalaya, Chhattisgarh, Tamil Nadu, Jharkhand, Uttar Pradesh, Madhya Pradesh, Gujarat, Kerala, Maharashtra, Jammu Kashmir, Punjab, Haryana, West Bengal, Bihar, Assam.

We use Twitter Search API to obtain the twitter data. We pick the relevant attributes like tweet text, tweet location, search term, username etc from the data. We wanted to obtain tweets which are tweeted from a city that contain information about a state. We call this api with the latitude, longitude and radius of a city and a state (choosen randomly from the concerned set of cities and states) and obtain a number of tweets corresponding to them. We make multiple such calls to collect data for all city and state pairs.

We collect data using python scripts. We host these scripts on Google Collaboratory and store the code files as well as data on a Google drive dedicated to this project. This was done

to ensure smooth pipeline between the data collection and analysis works. Due to a common script, we can keep on collecting the data and obtaining results simultaneously on the updated dataset. The script is run at a fixed time every week for a fixed duration to obtain new data along with obtaining results on the updated dataset.

The data collection for this project has been done in 2 phases. The first phase was between September, 2019 - December, 2019 and the second phase is currently ongoing having started in February, 2021. The tweets collected are then filtered. The number of relevant tweets collected over the 2 phases are 78,191 and 54,034 (as of now), respectively making a total of 132,225 tweets.

4 Methodology

After collecting the data we preprocess it. The preprocessing step includes stop words removal, lemmatization etc.

Post preprocessing, we use 2 different methods to obtain the results from the data. The first method involves using TF-IDF vectorizer. The second method uses the doctag2vec model.

4.1 DocTag2Vec

DocTag2Vec is an extension of Word2Vec and Doc2Vec models. In DocTag2Vec model, we simultaneously learn the representation of words, documents, and tags in a joint vector space during training. Post training we obtain a vector corresponding to each document and each tag in a common vector space [5].

We model our entire corpus as set of documents with each tweet representing a unique document. We also tag each tweet with the city from which the tweet has been tweeted and the state to which the tweet is made. Thus each tweet has 2 tags. After training the model, we learn the representation of each tweet as well as tag in our vector space. We decide to obtain the vectors in a 25 dimensional vector space. The choice of number of dimensions was done with the following considerations : A larger number would be computationally heavy as well as can cause over-fitting at the same time. On the other hand, a low dimensional space would mean loss of information from the data.

Thus, in other words, post training we obtain a vector for each tweet as well as each city and each state in a common 25 dimensional vector space. This space contains all the information required to explore the relationship between different cities and state and therefore we continue our analysis.

Now, we cluster all the tweets using the unique city-state pair they are tagged with. For all the tweets belonging to a city-state pair, we find a unique representational vector corresponding to that city-state pair. We use the centroid of the cluster of tweets to represent the cluster. This vector represents the relationship of that city-state in our 25-dimensional universe.

Post obtaining these vectors we develop different models to find out the strength of migratory relationship between the corresponding city-state pair. We discuss these models and the results obtained in detail in the next section.

4.2 TF-IDF

For each unique city-state pair, the corpus contains all tweets corresponding to the city-state and each tweet is considered as a separate document.

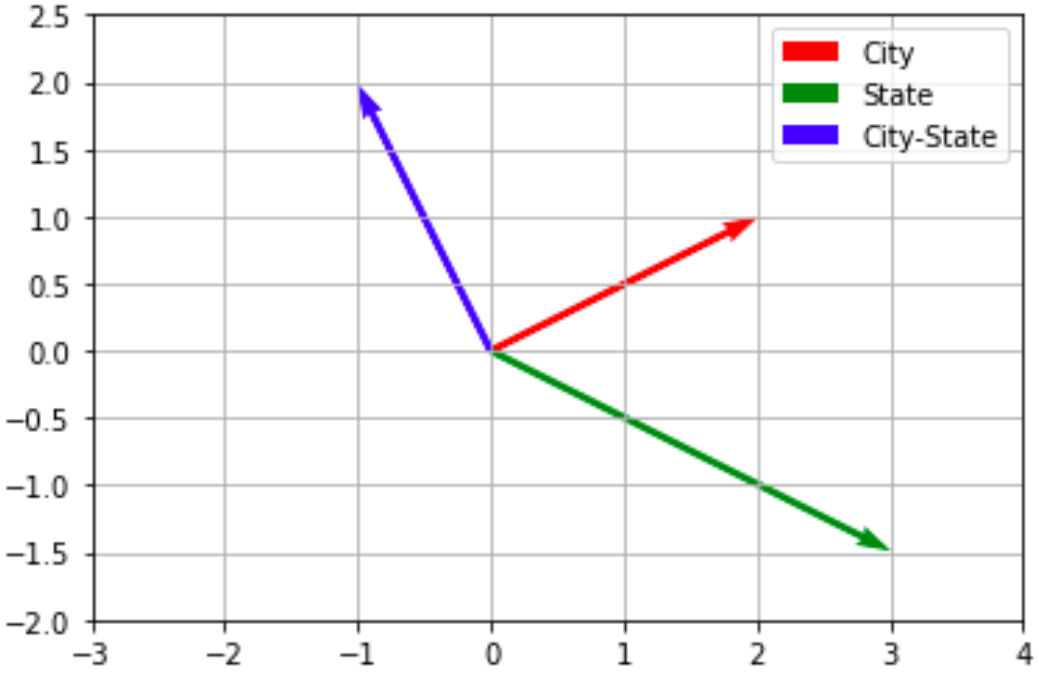

Figure 2: Vectors we obtain after Clustering in Doc-Tag 2 Vec (Compressed in 2-D)

On applying TF-IDF, we get scores for each word with respect to its relevance in the tweet. We combine the scores of words across all documents (tweets) in the entire corpus. Thus we obtain a large vector representing the relationship between the city-state pair. We get such vectors for all city-state pairs.

Our TF-IDF model, discussed in the next section, is based on obtaining the strength of the relationship between a city-state using the corresponding vectors.

	Rajasthan	Odisha	Tripura
New Delhi	[0.0, 0.0020379021921764283, 0.0, 0.0, 0.0, 0....	[0.0, 0.0015845955007805941, 0.0, 0.0, 0.0, 0....	[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, ...
Hyderabad	[0.0, 0.016305120055099948, 0.0, 0.0, 0.0, 0.0...	[0.0, 0.01666811694783955, 0.00092790035753127...	[0.0, 0.014762516287924491, 0.0, 0.0, 0.0, 0.0...
Kolkata	[0.0, 0.016473897050958922, 0.0, 0.0, 0.0, 0.0...	[0.0, 0.013725238856943203, 0.0, 0.0, 0.0, 0.0...	[0.0, 0.009661200845500532, 0.0, 0.0, 0.0, 0.0...

Figure 3: Sample score matrix obtained post TF-IDF

4.3 Results and Comparison

We obtain results in a common format for all our models. We basically derive a doubly stochastic matrix of size 18x18, for each model. Doubly stochastic means that all rows and columns of the matrix sum up to 1. The rows represent the cities while the columns represent the states.

To obtain a doubly stochastic matrix corresponding to our matrix, we use Sinkhorn Normal Decomposition. Sinkhorn in his paper showed that corresponding to each positive square matrix A, there is a unique doubly stochastic matrix $S = D_1.A.D_2$, where the D_1 and D_2 are diagonal matrices with positive diagonals. This doubly stochastic matrix, S, can be obtained as the limit of the iteration defined by alternately normalizing the rows and columns of A[6]. Thus using this idea, we obtain the corresponding doubly stochastic matrix for each of our result matrices and hence, compare them.

Thus, cell (i, j) can be interpreted as the probability of a migrant belonging to state j, migrating to city i given he/she migrates to one of the 18 cities. Similarly, it can also be interpreted as a migrant encountered in city i, belongs to state j, given he/she belongs to one of the 18 states.

We also obtain an expected matrix using the census data and apply the Sinkhorn Normal Decomposition to it and then, check the closeness of each of the matrices obtained in each of the models to this expected matrix.

5 Models: Ideas and Results

5.1 Model 1 (based on DocTag2Vec)

The idea is to compare the norm of each city-state representative vector. As this is the complete representative of the relationship of the corresponding city-state in our 25 dimensional world, it makes sense to expect that the norm of this vector would represent the strength of relationship of the concerned pair.

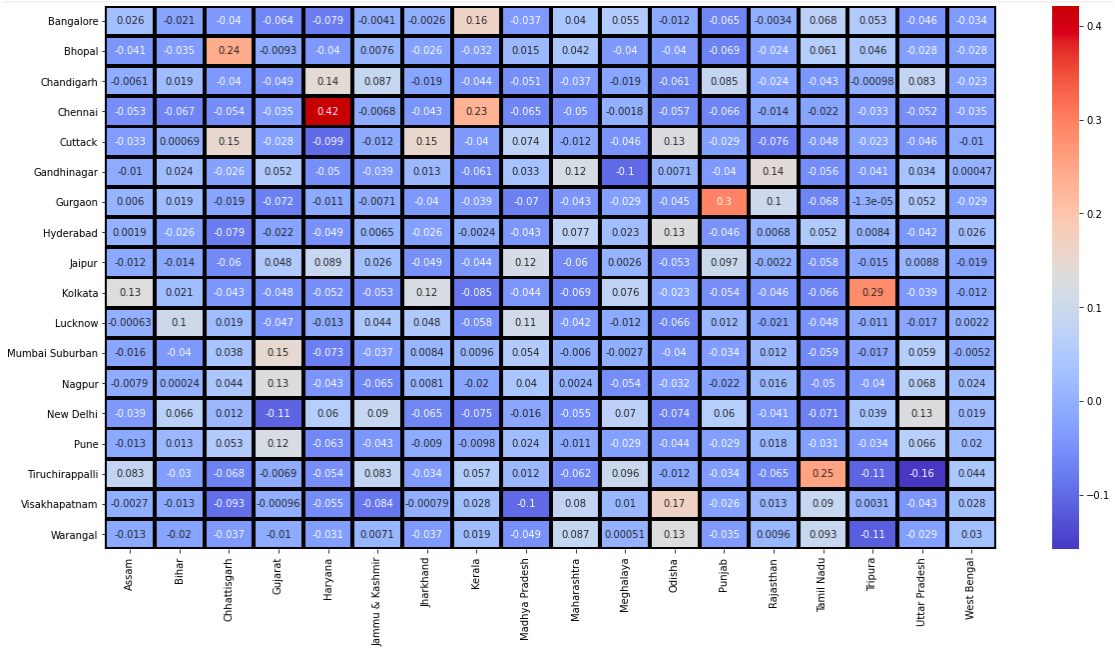

Figure 4: Model 1 Results

5.2 Model 2 (based on DocTag2Vec)

In the second model, we claim that the entire relationship between the city-state is not responsible for migration, but only a component of it is responsible.

To find this component, we try to define a new direction and then compare the norms of the components of each vector along this new direction.

The new direction is defined using the tag vectors (vectors for each city and each state). We define $\mathbf{C} = \Sigma \mathbf{c}_i$ (where $\mathbf{c}_i$ are the individual city vectors) as a super city, formed by resultant of addition of all city vectors. Similarly, we define $\mathbf{S} = \Sigma \mathbf{s}_i$ (where $\mathbf{s}_i$ are the individual state vectors) as a super state, formed by the resultant of all state vectors. Now the difference between $\mathbf{C}$ and $\mathbf{S}$, i.e. $\mathbf{C}-\mathbf{S}$, is the difference between the super city and super state of India, and thus the migration is happening because of a difference captured by the direction of this vector.

Thus, we find the component of each city-state vector along this vector and compare their norms.

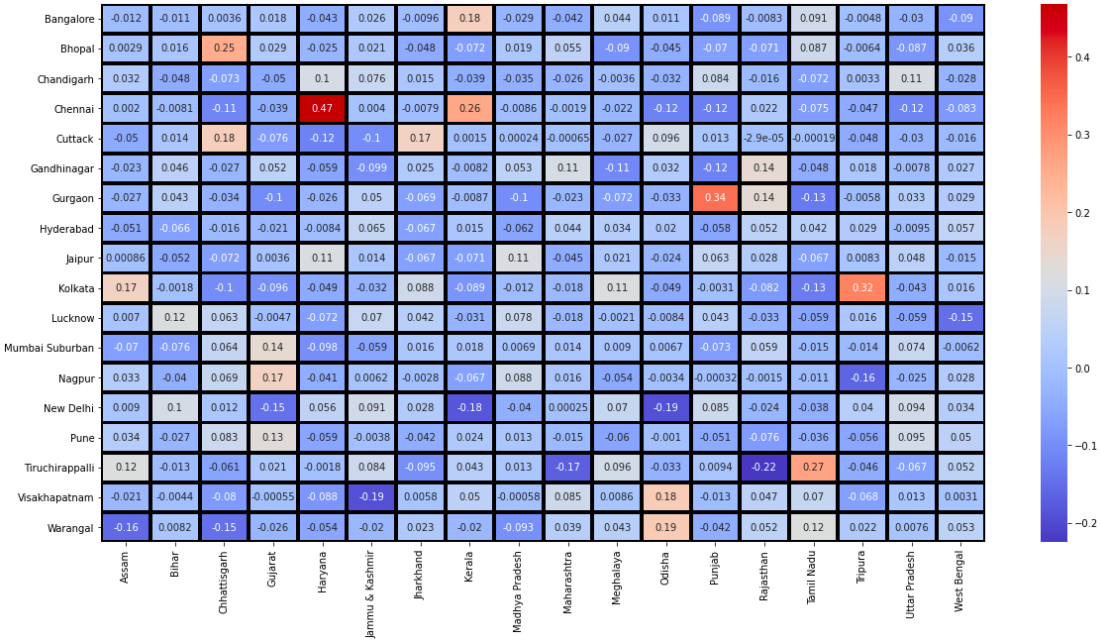

Figure 5: Model 2 Results

5.3 Model 3 (based on DocTag2Vec)

This is closely related to the idea of the previous model, but here the definition of $\mathbf{C}$ and $\mathbf{S}$, changes slightly. In order for each city and state to have an equal say in the super city and super state respectively, we form $\mathbf{C}$ and $\mathbf{S}$, not by adding the individual city and state vectors, but by adding unit vectors in direction of individual city vectors and state vectors.

$$\mathbf{C} = \Sigma \mathbf{c}_{i(unit)}, \mathbf{S} = \Sigma \mathbf{s}_{i(unit)}$$

Again, we find the norms of the components of each of the city-state vectors along C-S.

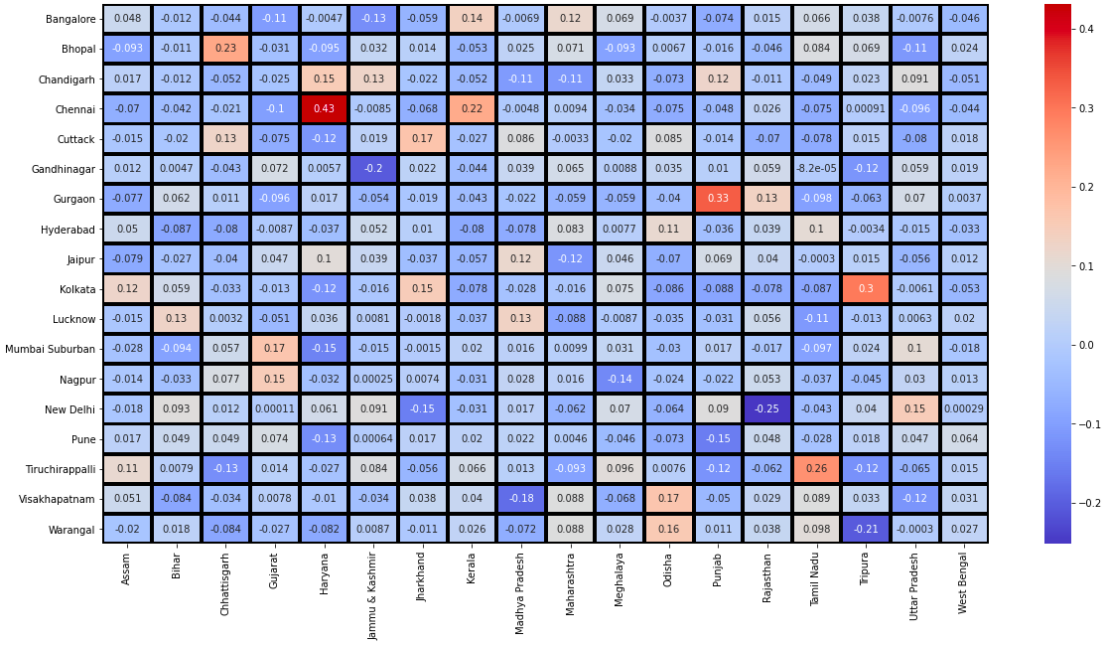

Figure 6: Model 3 Results

5.4 Model 4 (based on DocTag2Vec)

In the last two models, we try to find the component along the difference of super city and super state. Here we try something different. We claim that the migration between the city and state is mainly due to the differences between this particular city and state. Thus we find the norm of the component of the city-state vector along the corresponding (city minus state) vector.

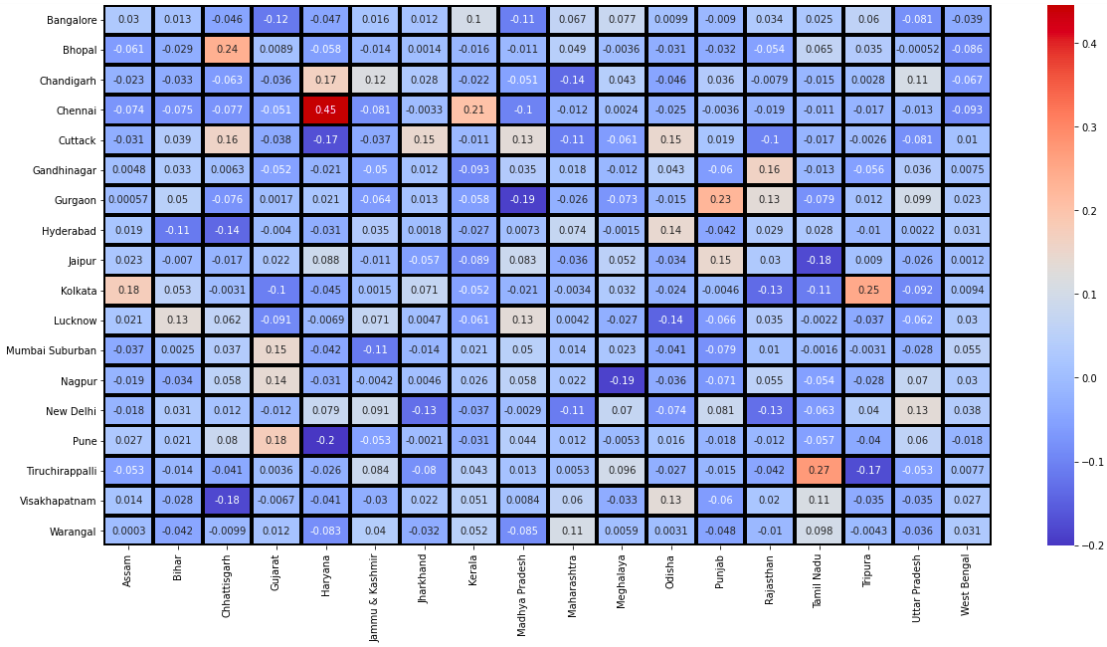

Figure 7: Model 4 Results

5.5 Model 5 (based on TF-IDF)

From the large word vector, obtained for each city-state vector, we pick the top 25 loaded words for each city-state pair. This forms a 25 dimensional vector for each of the city-state pairs. Now, this vector is claimed to be present in the same 25 dimensional world as the previous doctag2vec vectors with the axis rotated for different sets of vectors. Thus, we can compare the norm of this set of vectors with the doctag2vec vectors.

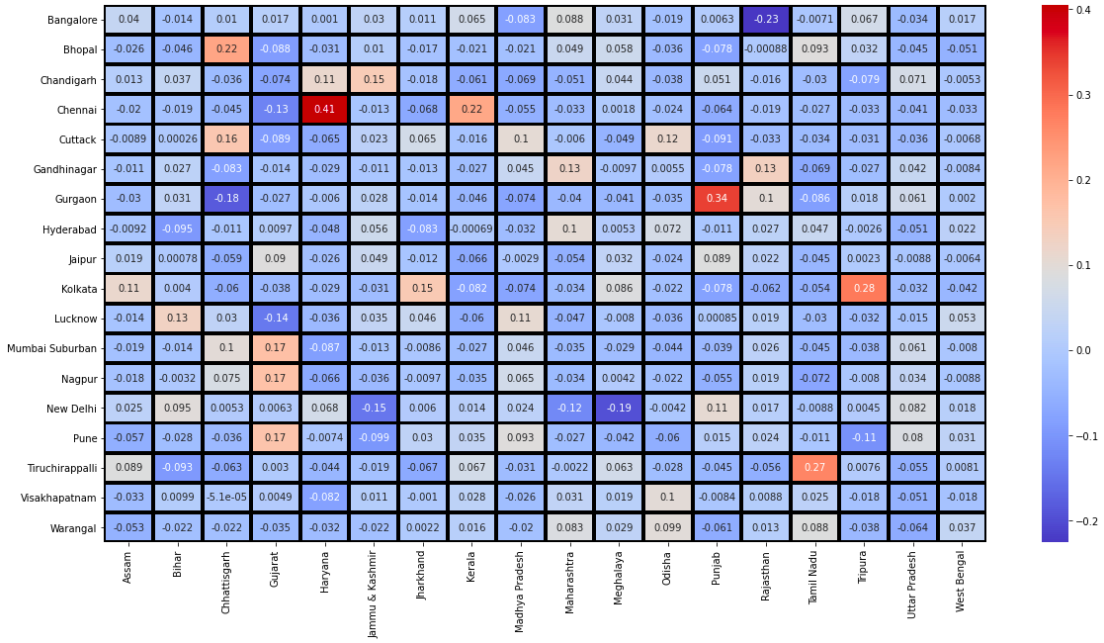

Figure 8: Model 5 Results

6 Comparison to Census Data

We obtain the following matrix using the Census 2011 data for corresponding cities and states. We perturb the intra state migration to the average value in the row, as we are primarily interested in the inter state migration. The matrix is then converted to a doubly stochastic matrix using Sinkhorn Normal Decomposition and is shown below :

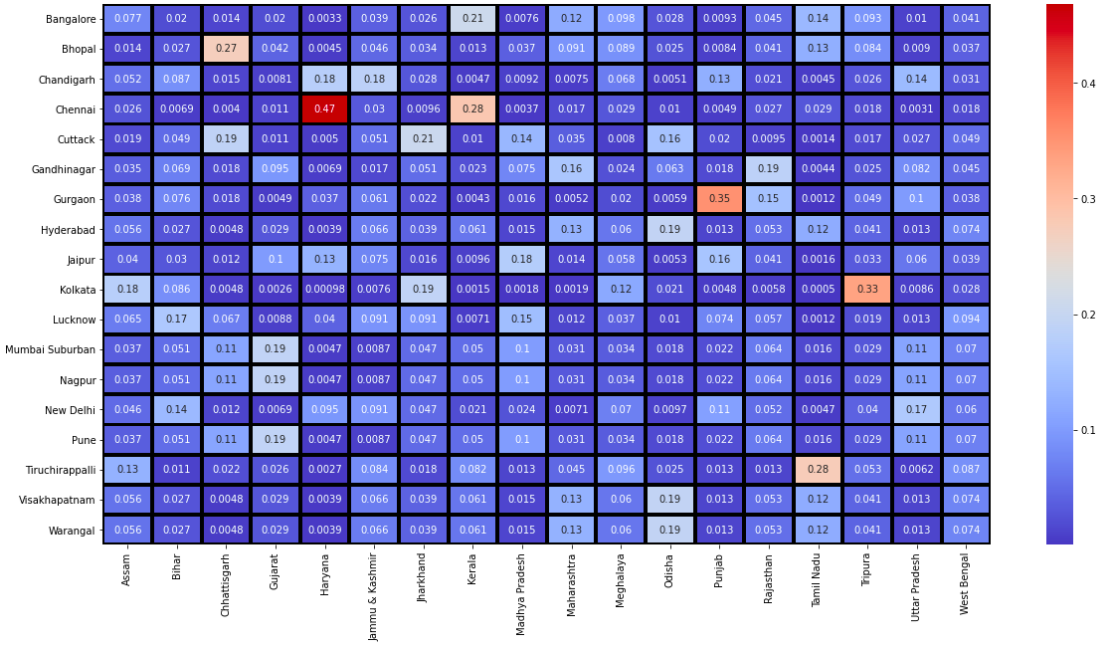

Figure 9: Census Data Matrix [1]

6.1 Frobenius norm

We define the distance between the matrices as the frobenius norm of the difference of the obtained matrix from the Census matrix. For a square doubly stochastic matrix with even rows and columns it can be proven that the maximum value of the frobenius norm can be $\sqrt{2n}$, where n is the number of rows in the matrix[8]. In our case $n = 18$, and thus the maximum value of frobenius norm can be 6. The minimum value of the frobenius norm can be 0.
We find the distance between the matrices obtained by various models from the census matrix and compare them. The lower the distance, the higher the closeness. We define a parameter closeness as $100 \cdot (1 - \frac{|F|}{6})$, where F is the Frobenius norm. The closeness parameter is defined such that in case of completely identical matrices, the value comes to be 100 (as $|F|$ would be 0), whereas for two matrices with maximum distance (6 in our case), the value will be 0.

6.1.1 Observations

The closeness of each model is as follows :

Model	Closeness
Model 1	79.38
Model 2	75.68
Model 3	75.78
Model 4	77.09
Model 5	78.80

Table 1: **Closeness of the Models**

Comparison : Model 1 > Model5 > Model4 > Model3 > Model2.
This comparison tells about which model is closer to the actual data in the sense of magnitude of Frobenius norm.

6.2 Rank Correlation

We now try to analyse the correlation between the rows and columns of the matrices obtained by our various models and the Census data matrix. We use the spearmans rank correlation for each row and each column of the 5 result matrices and census data matrix.

6.2.1 Row Correlation

For each model, we obtain spearmans rank correlation for each row with the corresponding row in the Census data matrix. This gives us an estimate as to how close This would give us an estimate of which model predicts a closer order of states with respect to that expected by the Census data. We list out below each row along with the model which has the highest correlation score with the Census data.

Model	**Best Model**
Bangalore	Model 5
Bhopal	Model 4
Chandigarh	Model 1
Chennai	Model 4
Cuttack	Model 2
Gandhinagar	Model 3
Gurgaon	Model 1
Hyderabad	Model 5
Jaipur	Model 3
Kolkata	Model 4
Lucknow	Model 2
Mumbai Suburban	Model 4
Nagpur	Model 2
New Delhi	Model 5
Pune	Model 2
Tiruchirapalli	Model 4
VisakhaPatnam	Model 5
Warangal	Model 5

Table 2: **Best Model for each of the Cities**

Model 1 works best for 2 cities, Model 2 for 4 cities, Model 3 for 2 cities, Model 4 for 5 cities, Model 5 for 5 cities

6.2.2 Column Correlation

For each modes result, we now use the same spearmans rank coefficient for finding the correlation between each column of the models result matrix with the Census data matrix. Again we find the model which worked best for each state and list the following below.

Model	Best Model
Assam	Model 5
Bihar	Model 4
Chattisgarh	Model 2
Gujarat	Model 4
Haryana	Model 2
Jammu Kashmir	Model 3
Jharkhand	Model 1
Kerala	Model 5
Madhya Pradesh	Model 3
Maharashtra	Model 4
Meghalaya	Model 2
Odisha	Model 4
Punjab	Model 2
Rajasthan	Model 5
TamilNadu	Model 2
Tripura	Model 4
Uttar Pradesh	Model 5
West Bengal	Model 5

Table 3: **Best Model for each of the States**

Model 1 works best for 1 state, Model 2 for 5 states, Model 3 for 2 state, Model 4 for 5 states, and Model 5 for 5 states.

6.2.3 Observations

In total, out of 36 rank comparisons done for each model, Model 1 worked best for 3 arrays, Model 2 worked best for 9 arrays, Model 3 worked best for 4 arrays, Model 4 and Model 5 worked best for 10 arrays.
Thus we can say that the models can be ranked in the following way (in order to explain the monotonicity of each row and column of result with the Census data):Model 4 = Model 5 > Model 2 > Model 3 > Model 1

6.3 Cellwise Deviation From the Census Data

For each model, we find the difference of the result matrix and the Census data matrix, and obtain a resulting matrix whose each cell is the value obtained by difference of values in the model result matrix and Census data matrix. After this, we find the patterns in the difference matrix.

6.3.1 Model 1

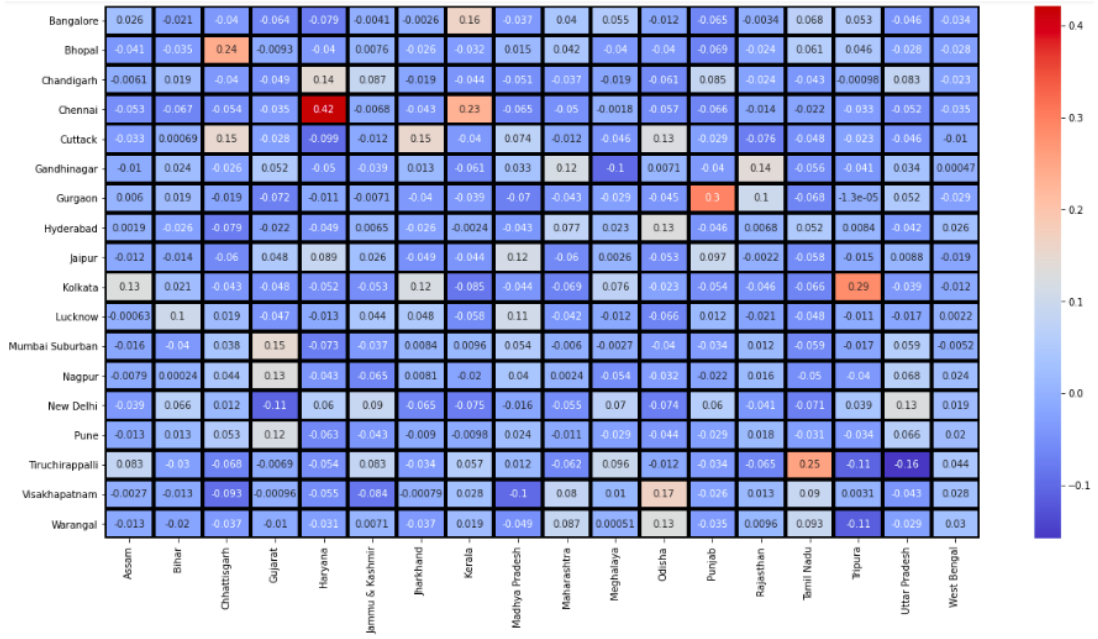

Figure 10: Model 1 difference from Census Data

City-State Pairs
Chennai - Haryana
Punjab - Gurgaon
Tripura - Kolkata
TamilNadu - Tiruchirapalli
Bhopal - Chattisgarh

Table 4: **City-State Pairs with the Top 5 differences**

Here 4 of the 5 entries are such that the city lies in the neighbouring state of the state from which migration is being concerned.

6.3.2 Model 2

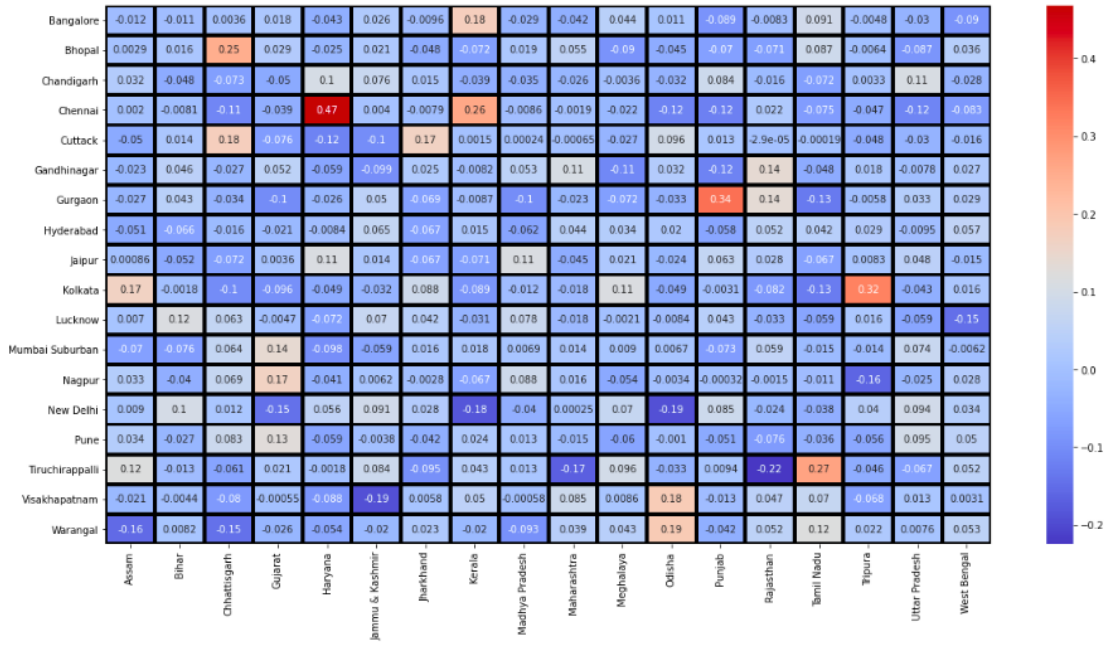

Figure 11: Model 2 difference from Census Data

City-State Pairs
Chennai - Haryana
Punjab - Gurgaon
Tripura - Kolkata
TamilNadu - Tiruchirapalli
Chennai - Kerala

Table 5: **City-State Pairs with the Top 5 differences**

Again, 4 of the 5 entries are such that the city lies in the neighbouring state of the state from which migration is being concerned.

6.3.3 Model 3

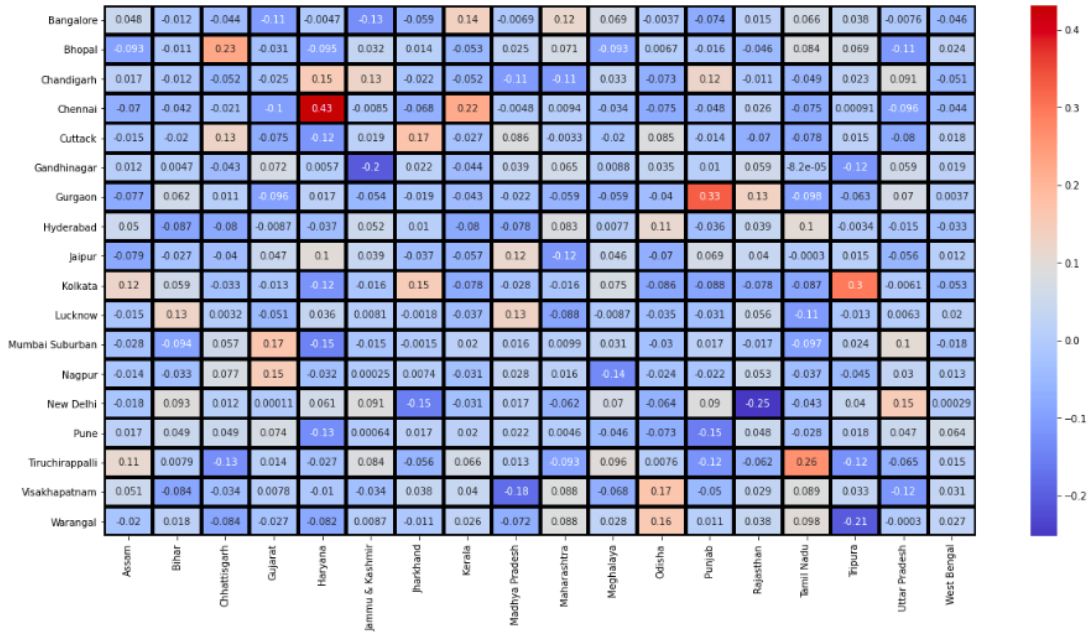

Figure 12: Model 3 difference from Census Data

City-State Pairs
Chennai - Haryana
Punjab - Gurgaon
Tripura - Kolkata
TamilNadu - Tiruchirapalli
Bhopal - Chattisgarh

Table 6: **City-State Pairs with the Top 5 differences**

Again, 4 of the 5 entries are such that the city lies in the neighbouring state of the state from which migration is being concerned.

6.3.4 Model 4

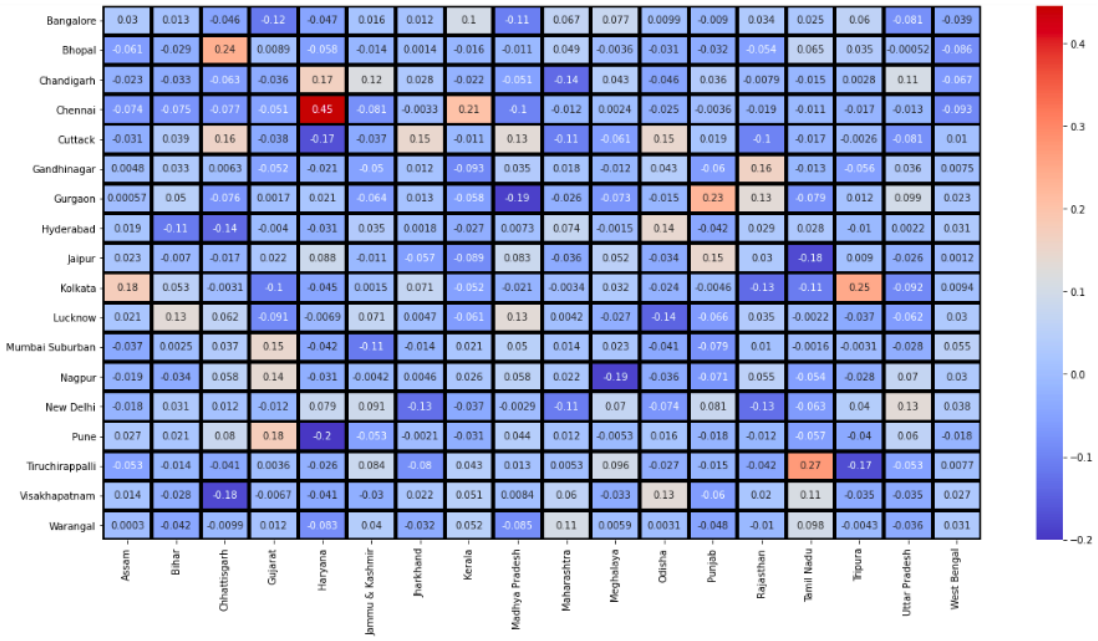

Figure 13: Model 4 difference from Census Data

States
Chennai - Haryana
Punjab - Gurgaon
Tripura - Kolkata
TamilNadu - Tiruchirapalli
Bhopal - Chattisgarh

Table 7: **City-State Pairs with the Top 5 differences**

Again, 4 of the 5 entries are such that the city lies in the neighbouring state of the state from which migration is being concerned.

6.3.5 Model 5

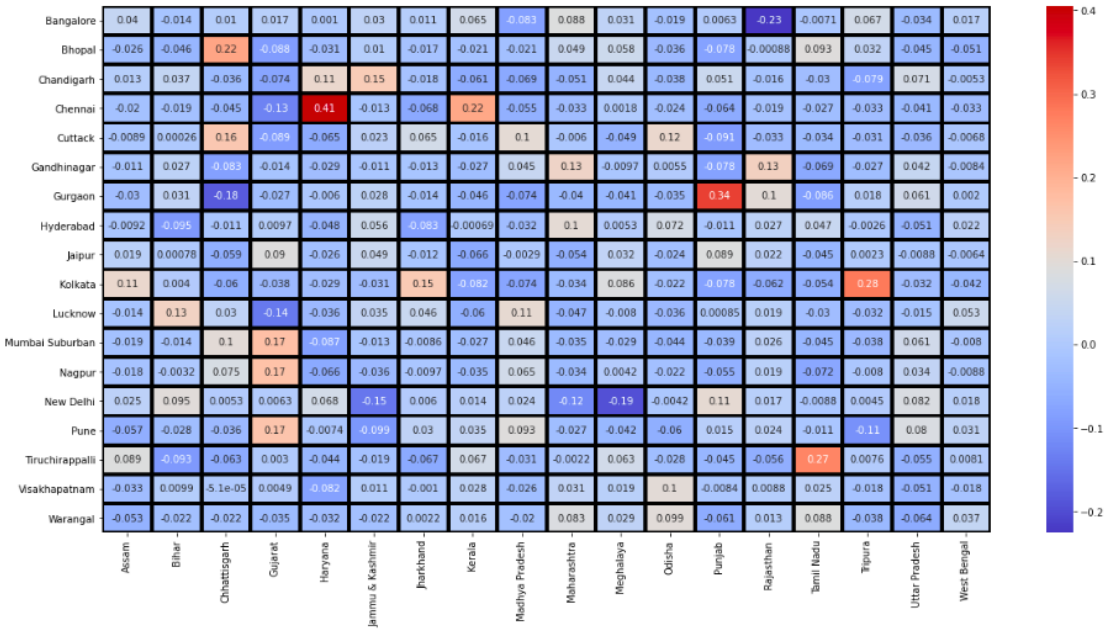

Figure 14: Model 5 difference from Census Data

City-State Pairs
Chennai - Haryana
Punjab - Gurgaon
Tripura - Kolkata
TamilNadu - Tiruchirapalli
Bhopal - Chattisgarh

Table 8: **City-State Pairs with the Top 5 differences**

Again, 4 of the 5 entries are such that the city lies in the neighbouring state of the state from which migration is being concerned.

6.3.6 Patterns Observed

In many cases it can be seen that the maximum difference occurs in city-state pairs where the geographical distance is low. A possible explanation can be that, the main population of migrants in census data are migrant workers, laborers etc who prefer to generally move to a tier1 city closer to their house. Whereas the twitter using migrants are generally salaried class of population whose movement is irrespective of geographical distance, and dependent on the opportunity they get in any city in India. This is due to more affordability of travel by the salaried class and more leave grants as compared to the daily wage workers. This causes a difference when we carry analysis of migration using twitter data and compare it to Census data.

7 Conclusion

We collect the twitter data in order to analyse the relationships between the cities and states in India. We construct multiple models for this analysis with different intuitions and ideas. We decided upon a unique format of result for each of our models as well as the verification data (Census based), i.e. a square doubly stochastic matrix. Thereafter, we compare the results of these models with the Census data using different methods like Frobenius norm, Spearmans correlation between rows and columns and deviation of each cell. We record certain observations for each of the methods of comparison.

References

[1] Census data on migration.

[2] Challenges when identifying migration from geo-located twitter data. 2021. c armstrong, a poorthuis, m zook, d ruths, t soehl. epj data science 10 (1), 1-14.

[3] Emerging patterns of migration streams in india: A state level analysis of 2011 census. 2018. r. ansary. migration letters.

[4] Mmovet15: A twitter dataset for extracting and analysing migration-movement data of the european migration crisis 2015. 2019. s urchs, l wendlinger, j mitrovi, m granitzer. 2019 ieee 28th international conference on enabling technologies: Infrastructure for collaborative enterprises (wetice).

[5] Sheng chen, akshay soni, aasish pappu and yashar mehdad. 2017. doctag2vec: An embedding based multi-label learning approach for document tagging. arxiv preprint arxiv:1707.04596.

[6] Sinkhorn, r., a relationship between arbitrary positive matrices and doubly stochastic matrices. ann. math. statist., 35 (1964), 876879.

[7] A study to assess pattern of migration across india based on census data. 2012. r taralekar, p waingankar, p thatkar. international journal of recent trends in science and technology.

[8] `https://math.stackexchange.com/questions/4154706/frobenius-norm-of-the-difference-of-two-doubly-stochastic-matrices`.

[9] Using face recognition with twitter data for the study of international migration. 2018. alexandru florea monica roman. informatica economica, academy of economic studies - bucharest, romania, vol. 22(4), pages 31-46.

[10] Smart phone users in india. `https://www.statista.com/statistics/467163/forecast-of-smartphone-users-in-india/`, March 2021.