EUROPEAN CONSORTIUM FOR MATHEMATICS IN INDUSTRY

ECMI Modelling Week 2018
University of Novi Sad, Faculty of Sciences

MODELLING HUMAN INTERACTIONS ACROSS A CITY WITH GRAPHS

Olivera Novović (BioSense Institute)

Lilla Lomoschitz	(Eötvös Loránd University)
Marc Monné Rius	(Autonomous University of Barcelona)
Maren Demuth	(Lund University)
Margarita Kan	(Saint Petersburg Polytechnic University)
Matthias Steinhausen	(Technical University Darmstadt)
Nemanja Filipovic	(University of Novi Sad)

December 10, 2018

Contents

1 Introduction

The analysis of telecommunication data enables interesting insight into human interactions inside and outside of cities. In the past years several studies have been performed for example on urban sensing, transport planning or multiple social analysis including epidemics or infectious diseases. In the current study anonymous telecommunication data for the city of Milan is used to extract patterns of human behavior inside and outside the city of Milan and capture the pulse of the city. The main focus of interest are the importance of city regions for human interactions and the temporal and spatial development during a day. Four distinct days are therefore analyzed. In particular, a normal workday, a day at the weekend and two days during the public holidays, one of them being the first day of Christmas. To obtain insights in human interaction graph theory is used to visualize and analyze the huge amount of telecommunication data within the inner city of Milan.

In the following, the provided data for this project and the main preprocessing steps are presented. This data is then used to analyze the overall traffic within the city of Milan. This insight provides the basis for a further investigation of human interaction using graph theory. Thereby, link significance and page rank are of major interest. The work concludes with a discussion of the obtained results and a short statement regarding the group dynamics during the project.

2 Data structure

Within this work anonymous mobile phone data for the city of Milan is used. To reduce the amount of data only four typical days in the city of Milan are analyzed:

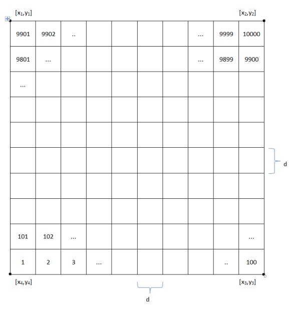
- A weekday (Thursday, 07.11.2013)
- A Sunday (Sunday, 01.12.2013)
- A holiday (Friday, 01.11.2013)
- First day of Christmas (Monday, 25.12.2013)

The provided telecommunication data is anonymized by *Telecom Italia* due to data protection regulations. The data is spatially and temporally discretized holding 10 minutes time frames for a 100x100 grid over the city of Milan. The grid divides the city area in small rectangles that hold a unique *SquareID* (SQID) with a fixed spacial position, see figure 1a. Figure 1b shows the grid overlayed with a city map of Milan. The telecommunication data provides information about the incoming and outgoing traffic over time between the SQIDs. Each data point consists of a timestamp, the origin of the traffic (SQID origin), the destination of the traffic (SQID destination) and the communication strength for the given time going from SQID origin to SQID destination, see table 1. The communication strength is given in Directional Interaction Strength (DIS) which is proportional to the volume of mobile traffic between two points. This includes the length and number of mobile calls, and the number of SMS-es sent.

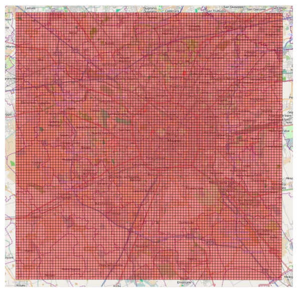
Originally, these informations are stored as Call Detail Records (CDR). Once a user is communication with his phone (SMS or call) a CDR is created in the telecommunication database. It contains multiple informations regarding, e.g. the time, duration, source and destination number and is used by the telecommunication provider to create the invoice. The CDRs used within this work are provided by the Semantics and Knowledge Innovation Lab (SKIL) of Telecom Italia (Barlacchi et al. 2015). Even though we cannot access the detailed information of the CDRs, as Telekom Italia did not want to share the exact data; the provided information is sufficient for our purposes, as we are only interested in the relation between the city regions.

Table 1: Example data points of the telecommunication data for Friday, 01.11.2013

Time	SQID origin	SQID destination	DIS
1383297600000	1	1	1.44E-4
1383300000000	1	1	2.89E-4
⋮	⋮	⋮	⋮
1383334800000	4017	3919	2.92E-4
1383346200000	4017	3919	3.88E-4



(a) Structure of the Milan grid



(b) Milan city map overlayed with the grid

Figure 1: Grid over Milan

2.1 Preprocessing of the data

Due to the temporal and spacial scaling the provided data consists of >50 million rows per day and needs to be to be preprocessed to reduce this huge amount of data. For the data aggregation the python library *pandas* was used. The preprocessing is done is several steps:

1. The data is filtered so it only contains data connected to the inner city regions of Milan. Traffic coming from and leaving to outside regions is ignored. Therefore, a dataset matching the city districts with the given SQIDs is used to assign an additional column to the original dataset holding the district number by a simple merge command.

2. The spacial resolution is reduced by using 88 districts of the city of Milan, taking the sum of all traffic over the SQIDs inside these districts. Therefore, the dataset was sorted and summed over all district numbers and times.

3. The temporal resolution is reduced, providing two data sets one summed over each hour and one over each day, respectively. For the temporal aggregation an additional time step was added to the data, transforming the original time step in the 10th minute and the hour the day, respectively. The data was then sorted and summed as for the spacial aggregation.

Table 2 displays the data structure after the temporal and spacial aggregation. Aggregation always comes with a loss of accuracy. Nevertheless, in order to analyze the overall telecommunication trends data aggregation is reasonable in order to get an representative overview of the hole city of Milan. Thereby, the data is condensed to major points of interests, making a data visualization and analysis possible. Figure 2 shows the districts of Milan in which the data is analyzed. Each node of the graph represents a district centroid and the edges with weights refer to the telecommunication traffic. A further explanation is done in later chapters.

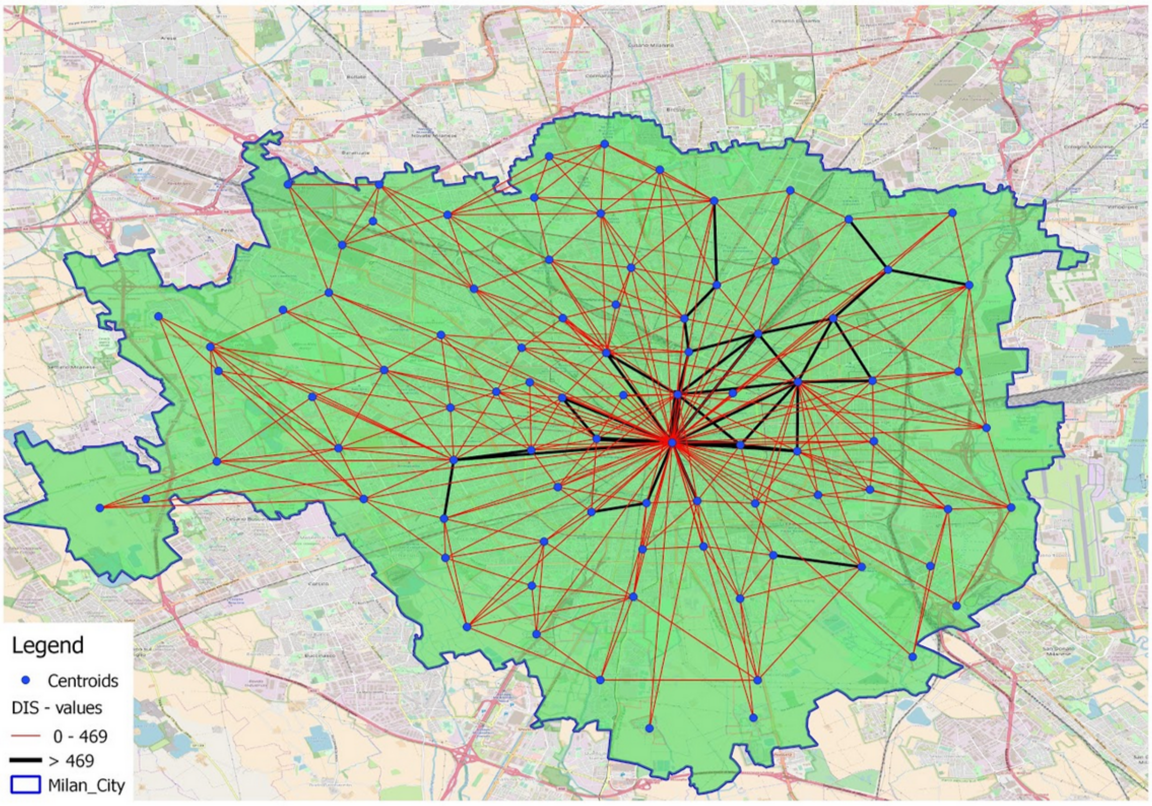


Figure 2: Graph of the communication inside the city of Milan, showing the different city districts as points.

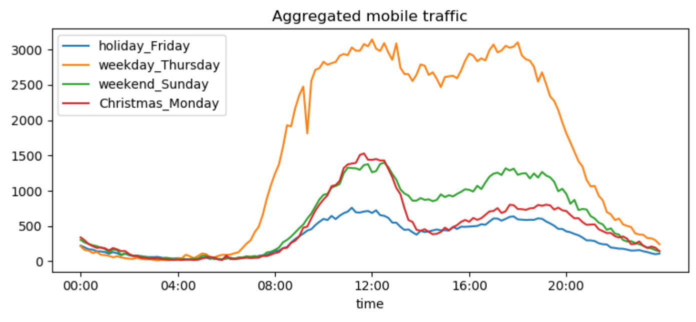
Table 2: Example data points of the telecommunication data

Time	District origin	District destination	DIS
1383297600000	1	1	5E-3
⋮	⋮	⋮	⋮
1383346200000	88	88	3.88E-3

3 Analysis of the overall traffic

As a first approach to gain insights the summed communication strength leaving from each district is analyzed. Therefore the telecommunication data is further condensed using the summed outgoing traffic for all city regions, no matter the direction. This provides first insights in interesting city regions and daily traffic development that can be used to pinpoint districts of interests for a further analysis using graph theory.

3.1 Temporal analysis



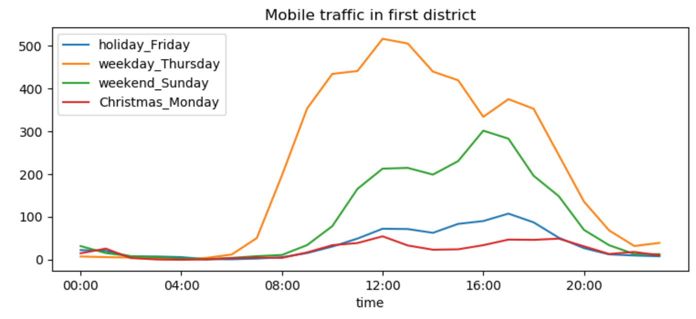


Figure 3: Mobile traffic throughout the four days

Figure 3 shows the overall traffic in the city of Milan and the traffic inside the city center close to the dome. Both subfigures include the traffic strength on the 4 different days. On the upper subfigure the data was aggregated into ten minutes intervals, while in the lower one it is aggregated by hours, as there the amount of data was less for only the city center, and for that the ten minute intervals showed more irregular peaks, which we considered to be too noisy.

The mobile communication strength on a workday is nearly twice as high compared to a holiday or the weekend, while the holiday and weekend do not differ significantly from one another. Additionally the first peak rising in traffic occurs one hour earlier on weekdays than on weekend or holiday. Despite the differences, daytime oscillations show several similarities: ups and downs occur at similar time intervals, until all days show a decrease at the end of the day. During nighttime, the communication strength is low for all four days.

Looking at the first significant rising of the mobile traffic, it could be assumed that this is the time when citizens wake up. For the weekday, the rising starts at 6 a.m., but becomes relevant between 7 a.m. and 8 a.m.. At 9 a.m., most of the traffic increment has been recorded, and the rising speed slows down until the traffic reaches its maximum at approximately 12 a.m.. For the other days, changes are not appreciable until 8 a.m.. The maximum of the mobile traffic stays at around 12 a.m., but the maximum traffic is smaller than for the weekday.

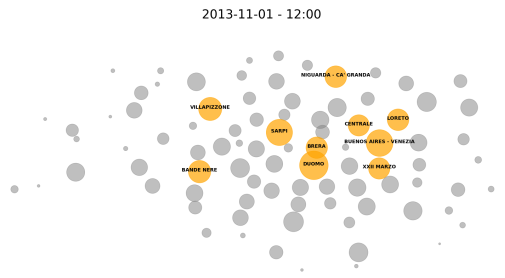
For the central city district the communication strength follows the overall traffic with some differences. The peak at 12 a.m. on the weekday is stronger, and the second peak in the afternoon is missing on the weekday. But for the Sunday it is the other way round, the peak at noon is smaller, but in the afternoon it is higher. The Friday holiday is quite similar to the aggregated data, but Christmas Monday differs a lot. Even though the communication strength for the first district is much lower compared to the whole city, the overall communication strength still is comparable to weekend Sunday.

3.2 Spatial analysis

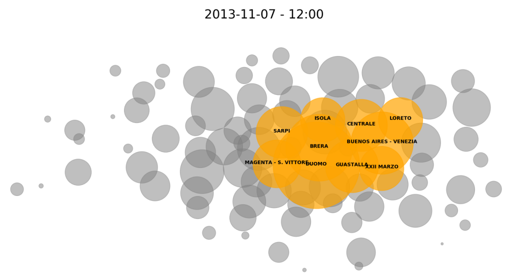
To further investigate the data an analysis of the city districts has been done. Figure 4 shows the communication strength at 12 a.m. for all four days. The radius of the district points are scaled according to the overall communication strength. Additionally the top 10 districts are highlighted in a orange color. While for the other days the communication hot-spots are in the city center, for Christmas Monday the communication strength is evenly distributed over the whole city. An explanation could be, that people in Milan celebrate the first day of Christmas at home or call their relatives which are spread throughout the whole city and do not all live in the first district Duomo.

3.3 Difference between in and outgoing traffic

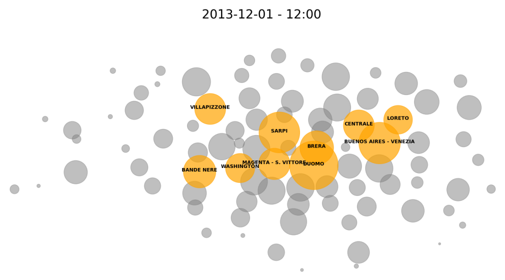
We were also interested if there is a difference in the pattern of the incoming and the outgoing traffic. So for each district we aggregated the data for the incoming and for the outgoing traffic through the whole days. For this the python library *pandas* was used. We grouped the data by the origin - destination district pairs. Then aggregated for each district all the traffic where it was the origin, so all outgoing traffic, and also all the data where that district was the destination, so all the incoming traffic. However we did not take into account the traffic staying inside the district for any of the above computations. The results of these data manipulations can be seen in Figure 5, so for these figures the traffic happening inside the districts is not included, only the communication which was happening between two different districts was used. It can



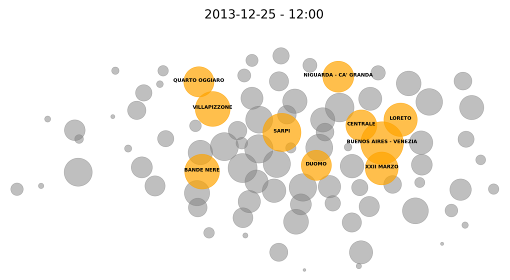
(a) Holiday Friday



(b) Weekday Thursday



(c) Weekend Sunday



(d) Christmas Monday

Figure 4: Comparison of the overall traffic in the districts of Milan at 12 a.m..

be seen from the pictures that most districts have approximately the same amount of outgoing and incoming traffic, but there are some exceptions. On the weekday district 80, and also district 1 had a significantly higher amount of outgoing traffic compared to the incoming, but this characteristic vanished for the Christmas holiday. The fact may be that there are more offices in this areas.

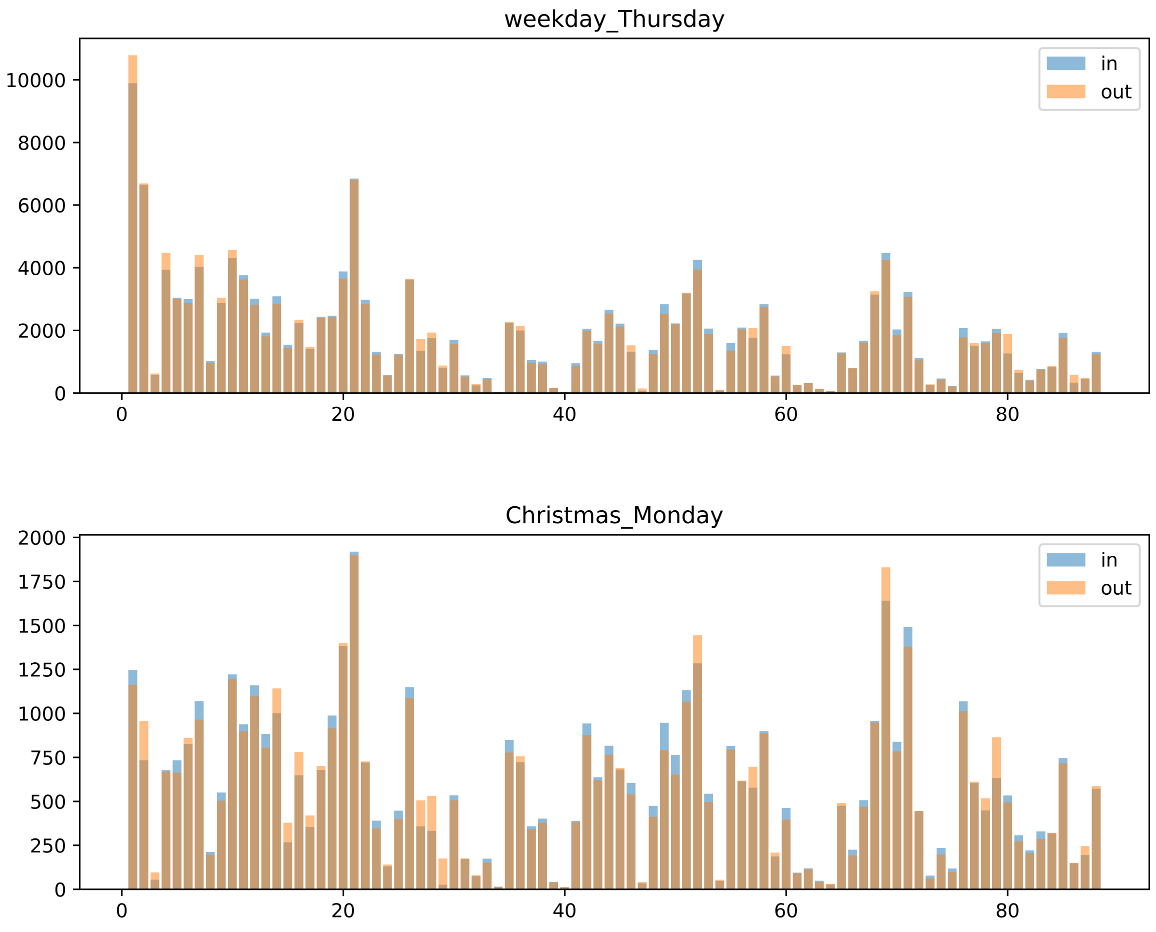


Figure 5: Difference between in and outgoing traffic

4 Graph theory application

4.1 Background

To analyze the data it is useful to apply graph theory. The districts of the city can be represented as nodes in a graph, while the links between them can be considered as edges. The natural planar representation of the graph can be easily achieved by taking the coordinates of the center of each district and linking them to the nodes. This way plotting the nodes at their corresponding coordinates will keep the shape of Milan city. Furthermore, edges from a graph can have weights, which can be used to contain the information about the DIS parameters that represent the strength of the links.

Originally, the graph should be directed because it makes difference whether the call is incoming or outgoing from the district. However, directed graphs show too much information when plotted which makes the viewer loose focus from the main results that can be extracted from the visualization of the graph. For that reason, most of the graphs used for visual purposes are undirected, while directed graphs are used just for analysis. To get the undirected graph from an originally directed graph, the sum of weights is taken as the new weight when both directions exist between two nodes.

4.1.1 Link significance

Graph theory application makes it easy to visualize the links between the districts but, since there are so many of them, it is important that only the most significant links are represented. To filter the edges the *link significance* parameter is introduced. This value depends on the number of nodes in the graph and on the probability of having a considered link. Equation (1) shows how link significance is calculated (Novović, Brdar, and Crnojević n.d.):

$$s_{ij} = 1 - (N-1)\int_{0}^{p_{ij}} (1-x)^{N-2}dx \tag{1}$$

where N denotes the number of nodes of the graph (for the city of Milan, $N = 88$: one for each city district), and p_{ij} is the probability of having link between nodes i and j.

Probabilities are calculated differently for directed or undirected graphs, according to equations (2) and (3) respectively. In these equations, dis_{ij} is the Directional Interaction Strength representing the traffic going from node i to node j. In the first case, probabilities are obtained by dividing dis_{ij} by the sum of all the DIS outgoing from district i which is not coming to district j, while a slight modification is implemented in the second case in order to compensate the fact that the direction of the traffic is no longer taken into account. Thus, in the second case, all the DIS between district i and any other district appears in the denominator:

$$p_{ij}^{directed} = \frac{dis_{ij}}{\sum_j dis_{ij} - dis_{ii}} \tag{2}$$

$$p_{ij}^{undirected} = \frac{dis_{ij}}{\sum_j dis_{ij} + \sum_j dis_{ji} - dis_{ii}} \tag{3}$$

In order to discriminate between strong links and weak links, the significance threshold α is applied in such a way that a link is considered strong or significant if $s_{ij} \geq \alpha$, being the weak links not represented in the graph.

4.1.2 Page Rank

Using basic graph properties, we can already obtain a lot of information on the activity between the districts. However, there are more advanced properties as well. One of them is the *Page Rank*.

Page Rank was introduced by Google for measuring the importance of a website. The idea is that if web page A links to page B that means that page A considers page B to be important. The importance of the page also reflects on its links importance: if a lot of important pages link to page B then the links of page B also become more important.

The formula used for iteratively calculating Page Rank is (Page et al. 1998):

$$PR(n_i) = \frac{1-d}{N} + d \sum_{p_j \in M_i} \frac{PR(p_j)}{L_j}, \tag{4}$$

where n_i is the i^{th} node, d the damping factor, N the number of nodes, M_i the set of nodes linking to n_i and L_j the total amount of outbound links of p_j. For a web site the damping factor d represents the probability of the situation when a user continues with another link from this page (and does not log out). Usually, d is taken as $0,85$. In our case Page Rank of a particular node represents likelihood of a random call to end up in the corresponding district.

4.2 Application and comparison of different days

We are now applying aforementioned graph properties to our data set and present results.

In figure 6 the undirected graphs for the four considered days are represented. The nodes are colored according to their Page Rank value: the greener a node, the bigger is its Page Rank value. The edges which are shown in the graphs are the ones that surpassed the significance threshold $\alpha = 0.05$. In addition, the 20 links with the highest DIS are represented by thicker edges than the others. As expected, these 20 edges tend to connect nodes with high Page Rank values.

As we can see from Figure 6, on an average working day (e.g. weekend Thursday) the nodes with the biggest Page Rank values are concentrated in the city while districts which are far from the city center do not have high Page Rank values.

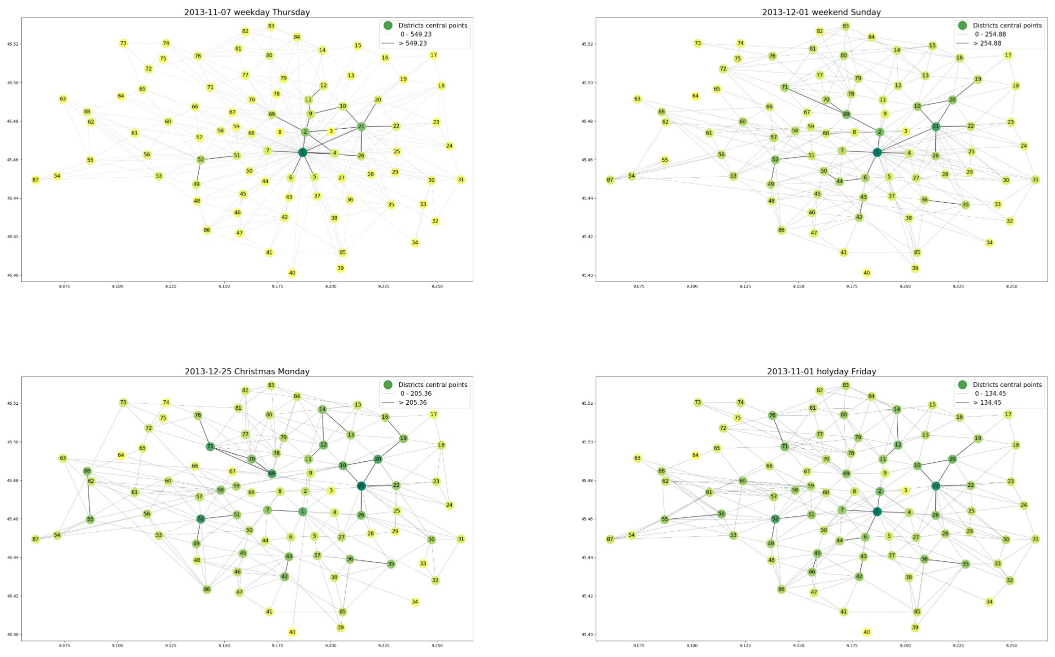


Figure 6: Page Rank Daily illustration

Duomo also remains the district with the biggest Page Rank value on holiday Friday and weekend Sunday but at the same time one can see that the activity spreads all around the city. As it follows from the graphs from previous sections, during Christmas Monday Duomo district is not the one with the biggest Page Rank value, though it is quite big still. Christmas Monday is also the day with the most widespread activity around the city.

Figure 7 represents hourly Page Rank for four different districts for one chosen day (weekend Friday). Looking at it we can conclude that during the day central districts are more likely to have incoming calls than districts which are far from the center.

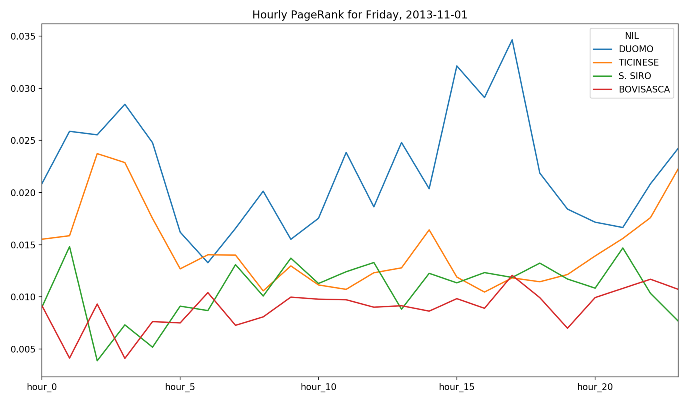


Figure 7: Hourly dynamics of Page Rank

Figure 8 represents variance in Page Rank per hour. Low variance implies more evenly distributed calls. As one can see, during the day the variance is low, while at night the variance becomes higher - that means calls which income to some particular districts. (These might be the ones where taxi offices, bars, pubs etc. are located).

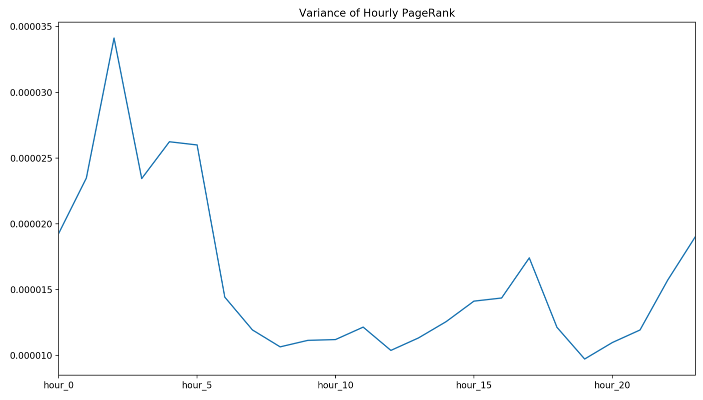
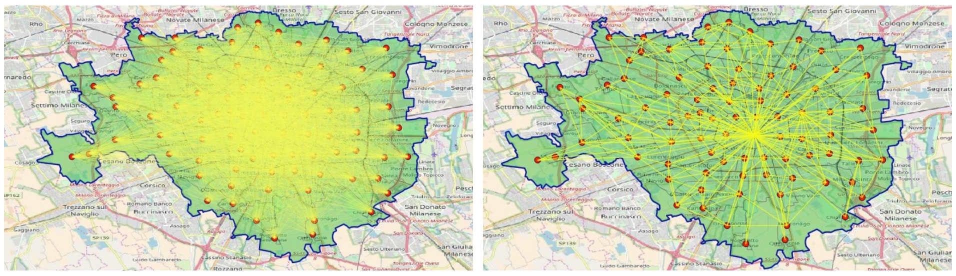


Figure 8: Hourly variance of Page Rank

4.3 Graph visualization

For visualization of graphs we have used NetworkX library in Python and QGIS, compare figure 9. The edges have been filtered using the link significance parameter.



(a) Before filtering.

(b) After filtering.

Figure 9: Visualization filtering in QGIS.

We created a function create_graph. That function works as follows. I prepares given data, getting coordinates from the nilzone.csv file and adding coordinates as the nodes of the graph. With this way our nodes (nodes being district centers) will be represented as fixed points in space and the shape of Milan will be preserved. The function also has a Boolean parameter whether the graph is supposed to be directed or undirected. Depending on that function corresponding edges are added between nodes. Nodes and edges were drawn by function draw_networkx.

5 Discussion and conclusion

We have set out to investigate human interactions across the city of Milan. It is the purpose of this section to reflect on our results and give an outlook for following investigations.

Our analysis is based on telecommunication data for four distinct days. For these days we have found that overall the humans of Milan communicate via telephone most on weekday Thursday, while holiday Friday and Christmas Monday show comparably low interaction strength between districts. At the same time, the weekday is also the day in which inner districts have a higher amount of interactions and the other days show more evenly spread communication. We deem these results to be reasonable as they confirm our intuition.

From our observations it is also possible to imply other properties of Milan's citizens, e.g. that they tend to get up earlier on the weekday than on the other days, since telecommunication interactions increase in strength earlier in the day. Also it might be that the people of Milan go out to bars and restaurants on Friday evenings where it is more likely that inner city districts (which typically contain the viral areas of town) are called. Note though that these statements have to be understood with caution, since we did not actually analyze sufficient data that would prove these claims, such as alarm clock settings or restaurant visits.

Indeed, we only investigated four days of data which sets a heavy limitation on the ability to generalize our results. In some occasions we also only compared selected districts. To state our results with higher confidence, a second iteration of our analysis can be carried out in which more days are investigated. Also, it could be interesting to not only include more days, but also enrich the data with different sources, e.g. include mobile data usage or social media interactions as well as GPS data.

To mention further limitations, we suffered from a lack of computational power which forced us to aggregate our data to districts and hours / days. This resulted in a loss of information on a respectively lower level. For example, if there were severe differences of telecommunication interactions in a district itself, we would not have detected that. This is another good point for future investigation.

Finally, it needs to be stated that we focused on data visualization using simple aggregation and graph theory. Though it is tempting to draw conclusions from that, we cannot imply causation in our analysis. We have however, keeping limitations and considerations in mind, successfully visualized and delivered insights into human interactions across the city of Milan.

6 Group work dynamics

The ECMI modeling week has been intense. We were challenged to tackle a complex problem together with people we had barely met the day before. And although we all came from different backgrounds, we were able to coordinate our work so that everybody delivered valuable results for the project.

First we focused on learning about data handling with Python and made sure that the project aim was clear to us, before we moved on to splitting the workload into smaller parts and assigned teams of two people to each part. These sub-projects were mostly claimed by interest and skill. To make sure that we kept the overview as a group, we had a daily gathering where we discussed the project status, possible problems or dead-ends and open questions. Overall the workload has been distributed evenly between all project members.

The classroom in Novi Sad made collaboration during the modeling week easy. Afterwards we had to replace the physical location by contact via mail and Slack. Since we organized ourselves well during the modeling week already, this worked out fine, with everybody writing about their sub-project in the report.

Beneficial about our group work setup was that we were able to quickly adapt workload according to our progress and shift resources between sub-teams when necessary. Our daily group meeting was an excellent source of creative ideas for further investigations and deep-dives into our project and visualization techniques, such as the animations and map-overlay we included in the final presentation. On the more challenging side we have to remark about the lack of computational power to handle such large datasets. This created unproductive wait times and the sheer size of the dataset bore the risk of not detecting errors in the data when handled incorrectly. We encountered no further challenges for which we could not find a solution.

All in all, we have been a well organized, motivated and independent group. Our diverse competencies allowed us to successfully complete the project, presentation and report.

7 Instructor's assessment

The ECMI modelling week is a great opportunity for networking and learning how to solve some practical problems. During the course of the week students have the opportunity to tackle some of the real world problems defined by the project. The title of the project assigned to the group is "Modelling human interactions across a city with graphs".

The data used for the project is real telecommunication data provided by Telecom Italia and it refers to spatial area of city of Milan. The data set consists of anonymized CDR records, which are further aggregated and analyzed in the context of areas with different spatial semantics.

The group was working very well as a team, and each student contributed individually to the project. The group was very interested in the subject they showed high ability to focus on the research problem. They paid good attention during the presentation of the problem and considered the explanations very carefully. They asked some really good questions which showed the overall deep understanding of the problem presented in the project and potential issues.

Dynamic of the work was well adjusted, the group had enough time to discuss the problem, embrace the practical implementation, and to give ideas and suggestions for future work. In the final stages they made some very good conclusions, and they showed the ability to interpret the analytical results in the context of geographical spatial semantic and human behavior patterns. In the final day, the group presented the methodology and the results very smooth and clear which shows undoubtedly the readiness to embrace even more complex problems.

A Code references

A.1 Data preprocessing

```
import networkx as nx
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# load data
select_sample = False
df = pd.read_csv('solution/data/MItoMI-2013-11-01_holyday_Friday.txt',
    sep='\t')
pointsdf = pd.read_csv('solution/data/points_in_zone_csv.csv')
df.columns = ['timestamp', 'SQID', 'sqid_destination', 'dis']

# just work with sample data
if select_sample == True:
  df = df.sample(frac=0.01)

# select df rows where sqids are in Milan
milan_sqids = pointsdf['SQID'].unique()
milandf = df[(df['sqid_destination'].isin(milan_sqids)) & (df['SQID'].
   isin(milan_sqids))]

# export to csv
milandf.to_csv('solution/data/processed/MItoMI-2013-11-01
   _holyday_Friday_milan.csv', sep='\t')
```

A.2 Temporal analysis

```
import os
import numpy as np
import pandas as pd
from shapely.geometry import Point
import matplotlib.pyplot as plt
os.chdir("C:\\path\to\files")

import glob
import datetime

files=glob.glob('processed/*relevantCityRegions.csv')
alldata_by_timestamps=[None] * 4
only_first_by_timestamps=[None] * 4
from_first_by_timestamps=[None] * 4
to_first_by_timestamps=[None] * 4
inner_by_timestamps=[None] * 4
outer_by_timestamps=[None] * 4
indexes=[None] * 4
i=0
for file in files: # DO SOMETHING (THE FILE DIRECTORY IS FILES)
    indexes[i]=file[28:][:-24]+'ay'

    milandf = pd.read_csv(file, sep='\t', names=['NIL_origin','
       NIL_destin','timestamp','dis'])

    min_time=min(milandf['timestamp'])
    milandf['timestamp_norm']=(milandf['timestamp']-min_time)
    milandf['hour'] = ((milandf['timestamp']-min_time+1000)
       /(1000*60*60)).astype(int)
    milandf['minute'] = ((milandf['timestamp']-min_time+1000)
       /(1000*60)).astype(int)%60

    milandf['time'] =  str(milandf['hour'])+":"+str(milandf['minute'])
    milandf_only_first=milandf[(milandf['NIL_origin'] == 1) & (milandf
       ['NIL_destin'] == 1)]
    milandf_from_first=milandf[(milandf['NIL_origin'] == 1)]
    milandf_to_first=milandf[(milandf['NIL_destin'] == 1)]
    milandf_inner=milandf[(milandf['NIL_origin'] == milandf['
       NIL_destin'])]
    milandf_outer=milandf[(milandf['NIL_origin'] != milandf['
       NIL_destin'])]

    grouped_by_timestamp = milandf['dis'].groupby([milandf['hour'],
       milandf['minute']])
    grouped_by_timestamp_only_first = milandf_only_first['dis'].
       groupby([milandf_only_first['hour']])#,milandf_only_first['
       minute']])
    grouped_by_timestamp_from_first = milandf_from_first['dis'].
       groupby([milandf_from_first['hour']])#,milandf_from_first['
       minute']])
    grouped_by_timestamp_to_first = milandf_to_first['dis'].groupby([
       milandf_to_first['hour']])#,milandf_to_first['minute']])
    grouped_by_timestamp_inner = milandf_inner['dis'].groupby([
       milandf_inner['hour'],milandf_inner['minute']])
    grouped_by_timestamp_outer = milandf_outer['dis'].groupby([
       milandf_outer['hour'],milandf_outer['minute']])

    alldata_by_timestamps[i]=grouped_by_timestamp.sum()
    only_first_by_timestamps[i]=grouped_by_timestamp_only_first.sum()
    from_first_by_timestamps[i]=grouped_by_timestamp_from_first.sum()
    to_first_by_timestamps[i]=grouped_by_timestamp_to_first.sum()
    inner_by_timestamps[i]=grouped_by_timestamp_inner.sum()
    outer_by_timestamps[i]=grouped_by_timestamp_outer.sum()

    i=i+1
```

A.3 Difference between in and outgoing traffic

```
import os
import numpy as np
import pandas as pd
from shapely.geometry import Point
import matplotlib.pyplot as plt
os.chdir("C:\\path\to\files")
import glob
files=glob.glob('processed/*summed.csv')
i=0
indexes=[None] * 4
out_sum=[None] * 4
```

```
in_sum=[None] * 4
inner=[None] * 4

for file in files: # DO SOMETHING (THE FILE DIRECTORY IS FILES)
    indexes[i]=file[28:][:-12]

    milandf = pd.read_csv(file, sep='\t', names=['NIL_origin','
       NIL_destin','dis'])

    milandf_out=milandf[milandf['NIL_origin']!=milandf['NIL_destin']]
    milandf_inner=milandf[milandf['NIL_origin']==milandf['NIL_destin'
       ]]

    grouped_by_orig = milandf_out['dis'].groupby(milandf_out['
       NIL_origin'])
    grouped_by_dest = milandf_out['dis'].groupby(milandf_out['
       NIL_destin'])

    out_sum[i]=grouped_by_orig.sum()
    in_sum[i]=grouped_by_dest.sum()
    inner[i]=milandf_inner

    i=i+1
```

A.4 Filtering connections by link significance

```
import pandas as pd
import numpy as np
import glob

files = glob.glob('data/summed/*')

for file in files:

  df = pd.read_csv(file, sep='\t', names=['NIL_origin','NIL_destin','
     dis'])

  for index, row in df.iterrows():
    if ((row['dis']!=0) & (row['NIL_origin']!=row['NIL_destin'])):
      symm = df[(df['NIL_origin']==row['NIL_destin'])&(row['NIL_origin
         ']==df['NIL_destin'])]
      if (symm.empty==False):
        df.at[index,'dis'] = df['dis'].iloc[index]+symm['dis'].iloc[0]
        df.loc[(df['NIL_origin'] == row['NIL_destin']) & (row['
           NIL_origin'] == df['NIL_destin']),'dis'] = 0

  df = df[df['dis']!=0.0]
  df = df.reset_index(drop=True)

  # define formula (not used any longer)
  def link_significance(p, k = 88):
    x = np.linspace(0,p,10000)
    y = lambda x: (1-x)**(k-2)
    y = [y(i) for i in x]
    return 1 - (k-1) * np.trapz(y,x)
  aij = lambda p: -1*(p-1)**87

  # calculate p value
  for k in range(1,89):
    print(k)
    inner_dis_value_for_k = float(df.loc[(df['NIL_origin'] == k)&(df['
       NIL_destin'] == k),'dis']) #remains the same
    disijs = df.loc[(df['NIL_origin'] == k) | (df['NIL_destin'] == k),
       'dis'].values
    js = df.loc[(df['NIL_origin'] == k),'NIL_destin'].values
    for j_idx,j in enumerate(js):
      df.loc[(df['NIL_origin'] == k)&(df['NIL_destin'] == j), 'p_value
         '] = disijs[j_idx] / (disijs.sum() - inner_dis_value_for_k)

  # calculate link significance
  df['link_significance'] = df['p_value'].apply(aij)

  # find strong and weak links
  df['link_strength'] = 'weak'
  df.loc[(df['link_significance'] < 0.05),'link_strength'] = 'strong'
  #TODO: find proper threshold for deciding on link strength

  # export file
  i = file.find('MItoMI-')
  j = file.find('_summed')
  file = 'data/processed/'+file[i:j]+'_link_significance_undirected.
     csv'
  print(file)
  df.to_csv(file, sep='\t')
```

A.5 Graphs visualization

A.5.1 GlobalGraph.py

```
import networkx as nx
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')

# DataFrame df should have at least NIL_origin, NIL_destin and
   link_significance columns
def changeDataFrame(df):
    df = df[(df['link_strength'] == 'strong') & (df['NIL_origin'] !=
       df['NIL_destin'])].sort_values(['dis'],ascending=False)
    df = df.reset_index(drop=True)
    return df

# DataFrame df should have at least NIL_origin, NIL_destin and
   link_significance columns
# Boolean if_directed shows if we want to get a directed or an
   undirected graph
# String path: path to the NILzone_centroids.csv file
# String date: the considered date
```

```
def createGraph(dataframe, path, if_directed, date, coef, num):

    # preparations
    df = changeDataFrame(dataframe)

    # getting coordinates of the future nodes
    coord = pd.read_csv(path, sep=',')
    coord = coord.sort_values(['ID_NIL'], ascending=True)
    coord = coord.reset_index(drop=True)

    # defining line thicknesses of the edges
    n = 30
    widest = 3.0

    if (coef == 0):
        coef = widest/df.at[0,'dis']

    lt1 = (df.at[0,'dis'])*coef
    lt2 = (df.at[n,'dis'])*coef/3
    df['line_thickness'] = lt2
    for i in range(0, n):
        df['line_thickness'].iloc[i] = lt1
    d = df[df['line_thickness'] == lt1]
    # creating the graph
    if (if_directed == True):
        G = nx.DiGraph()
    else:
        G = nx.Graph()

    # adding nodes
    for i in range(0, len(coord.axes[0])):
        G.add_node(coord['ID_NIL'][i], pos=(coord['XCOORD'][i], coord[
            'YCOORD'][i]))

    for i in range(0, len(df.axes[0])):
        G.add_edge(df['NIL_origin'].iloc[i], df['NIL_destin'].iloc[i],
            weight=df['line_thickness'].iloc[i])

    edgewidth = []
    for (u, v, d) in G.edges(data=True):
        edgewidth.append(d['weight'])

    if (if_directed == True):
        components = nx.strongly_connected_component_subgraphs(G)
    else:
        components = nx.connected_component_subgraphs(G)
    components = list(components)

    # getting nodes positions
    pos = nx.get_node_attributes(G, 'pos')

    # drawing the graph

    plt.subplots(1, figsize=(20, 20))
    nx.draw_networkx_nodes(G, pos, alpha=0.8, node_size=300,
       node_color=(34/255,139/255,34/255),linewidths=2, label='
       Central␣zone␣points')
    nx.draw_networkx_edges(G, pos, width=edgewidth, edge_color='k')

    proxies = [plt.Line2D([0,1],[0,1],marker='o',markersize=20,alpha
        =0.8,color="white",markeredgewidth=2,markeredgecolor
        =(34/255,139/255,34/255),markerfacecolor
        =(34/255,139/255,34/255)), plt.Line2D([0,1],[0,1],color='k',
        LineWidth=lt2), plt.Line2D([0,1],[0,1],color='k',LineWidth=lt1
        )]

    labels = ['Districts␣central␣points', '␣0␣-␣'+str(round(df['dis'].
        iloc[n],2)), '>␣' + str(round(df['dis'].iloc[n],2))]

    nx.draw_networkx_labels(G, pos, font_size=13, font_family='sans-
        serif')
    plt.legend(proxies,labels,fontsize=15)
    plt.rc('axes',titlesize=13)
    if (num == -1):
        time = ''
    else:
        time = '␣-␣'+str(num)+':00'
    plt.title(date+time,fontsize=20)
    plt.gca().set_xlim([(3*coord['XCOORD'].min()-9.075)/2,(3*coord['
        XCOORD'].max()-9.250)/2])
    plt.gca().set_ylim([(3*coord['YCOORD'].min()-45.41)/2,(3*coord['
        YCOORD'].max()-45.52)/2])
    mgr = plt.get_current_fig_manager()
    mgr.full_screen_toggle()
    date = date.replace("␣", "_")
    if (num == -1):
        tmp=''
    else:
        if (num<10):
            tmp = '0'+str(num)
        else:
            tmp = str(num)
    plt.savefig('plots/'+date+'_undirected_graph'+tmp+'.png',format='
        png',dpi=800)
    plt.close()
    return G, coef
```

A.5.2 Plotting.py

```
 import pandas as pd
import warnings
import GlobalGraph as GG
import importlib
import glob
importlib.reload(GG)
warnings.filterwarnings('ignore')

files = glob.glob('data/processed/*.csv')

path = 'data/NILzone_centroids.csv'

for file in files:
    coef = 0
    df = pd.read_csv(file, sep='\t')
    i = file.find('MItoMI-')
    i = i + len('MItoMI-')
    j = file.find('_link')
    date = file[i:j].replace("_", "␣")
    print(date)
    graph, coef = GG.createGraph(df, path, False, date, coef, -1)
```

A.5.3 Data for QGIS

```
df = pd.read_csv('path', sep='\t')
df = df[(df['link_strength']=='strong')&(df['NIL_origin']!=df['
   NIL_destin'])].sort_values(['dis'],ascending=False)
df = df.reset_index(drop=True)
df = df.drop(['Unnamed:␣0'],axis=1)
dfmerged = pd.merge(df,zone,left_on='NIL_origin',right_on = 'ID_NIL',
   how='left')
dfmerged1 = pd.merge(dfmerged,zone,left_on='NIL_destin',right_on = '
   ID_NIL', how='left',suffixes=['_from','_dest'])
dfreduced=dfmerged1.drop(['link_strength',],axis=1)
dfreduced=dfreduced[(dfreduced['NIL_origin'] != dfreduced['NIL_destin'
   ])]
strings=['LINESTRING'+'('+str(dfreduced['XCOORD_from'][i])+'␣'+str(
   dfreduced['YCOORD_from'][i])+','+str(dfreduced['XCOORD_dest'][i])+
   '␣'+str(dfreduced['YCOORD_dest'][i])+')' for i in range(dfreduced[
   'YCOORD_from'].shape[0])]
dfreduced['linestring']=strings
dfreduced.to_csv('path_for_QGIS.csv')
```

References

Barlacchi, Gianni et al. (2015). “A multi-source dataset of urban life in the city of Milan and the Province of Trentino”. In: *Scientific Data*.

Novović, Olivera, Sanja Brdar, and Vladimir Crnojević (n.d.). *Evolving connectivity graphs in mobile phone data*. University of Novi Sad.

Page, Lawrence et al. (1998). *The PageRank Citation Ranking: Bringing Order to the Web*. Stanford InfoLab.