Half a Mile, Half a World: Locality Patterns of International Calls in Milan

Francesco Malandrino
Politecnico di Torino
Torino, Italy
Email: malandrino@tlc.polito.it

Claudio Casetti
Politecnico di Torino
Torino, Italy
Email: casetti@tlc.polito.it

Carla-Fabiana Chiasserini
Politecnico di Torino
Torino, Italy
Email: chiasserini@tlc.polito.it

Abstract—**In a context of ever-increasing demand for network capacity, many scheduling algorithms and offloading techniques leverage content locality, i.e., the fact that nearby-located users tend to request the same content. In many cases, this phenomenon is linked to the fact that people with the same background, e.g., immigrants from the same country, tend to live close to each other. By exploiting real-world communication traces, we study the significance of such a clustering effect, and the extent to which it is linked to continent of origin of each community. We extend our study to Italian provinces, and see that, although similar to foreign communities in size, the people coming from a same province show no tendency to cluster together. We complete our paper with a discussion of the implementation techniques and programming models that are a better fit for such "big data" manipulation.**

I. Introduction

It has been often observed that humans react to relocation by seeking the familiar. Newcomer immigrants to London, for example, have been reported to either live *half a mile* from most of their fellow nationals, i.e., in the same neighbor or area, or go *half a world* back to their country of origin.

Recent data show that such a pattern is not true for modern immigrants; in particular, they tend to follow their job rather than their community [1]. However, immigration in London is an ancient and, in many ways, successful story; other cities, such as Milan, are widely believed to be following the same path. This prompts the question – how are immigrants distributed within Milan? Do they follow the "half a mile, half a world" pattern? Do immigrants from different areas, e.g., from different continents, behave differently?

In addition to foreign immigrants, Milan is home to a large number of immigrants from other parts of Italy. Do they show any tendency to cluster, or are such phenomena only associated to non-Italians, whose linguistic and cultural differences are more significant?

We do not ask such questions out of sheer curiosity. In addition to obvious economic and social implications, studying locality patterns of international communication has a direct impact on the operation of current and future cellular networks – and on their very profitability. Both are endangered by the raise in bandwidth demand [2], which also caused several recent connectivity shortages [3]. Among the many approaches that have been proposed to cope with this issue, most seek to leverage *content locality*, i.e., the fact that nearby-located users tend to request the same content. These users can thus be served through device-to-device LTE links [4], LTE broadcast [5], or opportunistic networking [6].

To network engineers, content locality is largely a fact, to be taken advantage of whenever possible. By studying the location of international callers, we seek to shed some light over its *causes*, making the phenomenon easier to understand, predict, and exploit.

Studying call locality patterns is difficult for several reasons. The foremost one is that all call information is confidential, collected by mobile operators and stored in deep vaults, inaccessible to ordinary researchers. Thankfully, Telecom Italia decided to share a valuable set of traces with the participants of the Big Data Challenge [7] they launched in 2014.

As essential as it is, obtaining the traces is but a first step. The next ones include identifying the meaningful information, processing it efficiently, and make sense of the result. There is also another, more technical challenge: "big data" such as our traces are, well, big: dealing with them requires special care, optimized tools, and tailored algorithms.

We begin by describing our traces in Sec. II. Then, we present the metrics we compute in Sec. III, along with the relevant implementation details. We discuss the results we obtain in Sec. IV, and draw our conclusions in Sec. V.

II. The traces

A rich set of traces, covering the months of November and December 2013 and the Italian cities of Milan and Trento, have been shared by Telecom Italia with the participants to its Big Data Challenge [7].

Space is divided into 10,000 235-meter tiles; time is divided into 10-minute periods. For each tile and time period, the traces contain a value proportional to the number of calls and SMS messages send and received, as well as Internet activity. Notice that we do not know the actual number of messages, duration of calls, amount of Internet traffic; this is due to privacy (and, possibly, commercial) reasons.

Information is further disaggregated by international prefix and Italian province. As an example, we can know whether there have been incoming calls from the province of Palermo to Piazza Duomo on Wednesday, Dec. 18th between 6:30 and 6:40, or on the same day between 19:10 and 19:20. Such level of detail has an impact on the size of traces: the full set of

TABLE I
Telecommunications_MI TRACE: FIELDS.

Field	Description	Type
Square id	Tile number (1 to 10,000)	Numeric
Province	Name of the province	Text
Time interval	Start of the 10-minute period of interest	Timestamp
Incoming calls	Value proportional to the number of incoming calls	Numeric
Outgoing calls	Value proportional to the number of outgoing calls	Numeric

TABLE II
Telecommunications_MI_to_Provinces TRACE: FIELDS.

Field	Description	Type
Square id	Tile number (1 to 10,000)	Numeric
Time interval	Start of the 10-minute period of interest	Timestamp
Country code	International prefix of the country	Numeric
SMS-in	Value proportional to the number SMS messages from the country to the tile	Numeric
SMS-out	Value proportional to the number SMS messages from the tile to the country	Numeric
Call-in	Value proportional to the number calls from the country to the tile	Numeric
Call-out	Value proportional to the number calls messages from the tile to the country	Numeric
Internet	Value proportional to the amount of Internet traffic generated from roaming phones of the country	Numeric

compressed archives weighs around 190 GByte and includes tens of billions of rows.

In addition to these communication traces, contestants have been given access to a significant amount of additional information, including road traffic, weather conditions, and geotagged tweets. These auxiliary traces share the same time and space reference of the main one, making it remarkably easy to correlate them.

Traces are distributed as compressed archives, containing a file (typically in TSV format) for each day.

In this work, we will be using only two of the traces, named *Telecommunications_MI* and *Telecommunications_MI_to_Provinces*. The information they include is shown in Tab. I and II respectively.

III. Metrics and implementation details

We start this section by describing and defining the metrics we are interested into (Sec. III-A). Then, in Sec. III-B, we provide further details on how to efficiently compute such metrics and process the traces.

A. Metrics

Ideally, we would like to know the average distance between two nationals of a given country (or two persons coming from the same province) and living in Milan – this would enable us to directly verify the "half a mile or half a world" pattern. Clearly, computing such a metric is unfeasible with the data at our disposal.

What we can compute is the average distance between people *calling* the same nation or province. Therefore, our first problem is to infer where a person lives from where she makes (or receives) calls. We cope with this issue by restricting our attention to the calls that take place after 8pm, and assume that (most) of such calls are made from one's house. Also, most likely these are personal, rather than business, calls. As simple as they are, such an assumptions are not overly simplistic: while it is true that one makes plenty of calls, e.g., on a night out, the calls we are looking at are international (or to another province) – this excludes, e.g., all calls made to a friend to know whether she wants to hang out that very night.

Let $\mathcal{T}$ be the set of all tiles, and $\mathcal{C}$ the set of all countries. We call p_{ct} the amount of (relevant) calls made between country $c \in \mathcal{C}$ and tile $t \in \mathcal{T}$ in either Milan or Trento. Also, let d_{uv} be the distance between tiles u and v.

We begin by estimating the fraction f_{ct} of people from country c that are in tile t at the time the traces are recorded. As discussed above, we do so by exploiting the amount of calls p_{ct}:

$$f_{ct} = \frac{p_{ct}}{\sum_{u\in\mathcal{T}} p_{cu}}. \tag{III.1}$$

Now, we can compute the average distance δ_{ct} at which a person from country c staying in tile t will find her fellow nationals:

$$\delta_{ct} = \sum_{u\in\mathcal{T}} f_{cu} d_{tu}. \tag{III.2}$$

Finally, the average distance $\bar{\delta}_c$ between nationals of country c can simply be computed through a weighted sum:

$$\bar{\delta}_c = \sum_{t\in\mathcal{T}} p_{ct} \delta_{ct}. \tag{III.3}$$

It is important to stress the meaning of the metric $\bar{\delta}_c$: *if* the p_{ct} values were the number of people from country c in tile t, *then* $\bar{\delta}_c$ would represent the average distance between two such people. In practice, we estimate the number of people through the calls they make; nevertheless, $\bar{\delta}_c$ is a good measure of how much communities tend to cluster together.

Finally, we stress that the average distance between people making calls to/from Italian provinces is computed in exactly the same way.

B. Implementation details

Having defined the metrics, one may think most of the job is done – we only have to implement the equations in any language of our choice, and feed the data to the resulting program. Sadly, when it comes to big data, *how* algorithms are implemented becomes a crucial issue. It is not merely a matter of doing operations faster – selecting the right tools impacts the scale of problems one can solve. Efficient implementation does not mean selecting "fast" languages, such as C/C++, but rather using optimized libraries and efficient paradigms, whichever the language.

TABLE III
OUTPUT OF THE *map* STEP.

Field	Description	Type
`Square id`	Tile number (1 to 10,000)	Numeric
`Country`	Name of the country	Text
`Traffic`	Number of incoming/outgoing calls	Numeric

1) Vectorized computation: Numpy [8] is a Python library aimed at scientific computing. It provides a `ndarray` type to represent multidimensional arrays (including matrices), and a highly optimized implementation of array operations (including matrix product). Such a power comes at a cost in terms of flexibility: in order to profit by Numpy's speed, operations must be *vectorized*, i.e., expressed in matrix form.

Our next task is thus to compute the metric $\bar{\delta}_c$ through matrix operations alone. Notice that such a need for vectorized operations is not exclusive to Python and Numpy: R and MATLAB users face the same issue.

We group the p-values in matrix $P = (p_{ct})$, and the d-values in matrix $D = (d_{uv})$. Then, we proceed as shown in Alg. 1.

Algorithm 1 Vectorized computation of the metric.

```
Require: $P, D$
1: $F \leftarrow P/P$.sum(axis=1)
2: $\Delta = (\delta_{ct}) \leftarrow FD$
3: $\bar{\delta} = (\bar{\delta}_c) \leftarrow (\Delta \cdot F)$.sum(axis=1)
4: return $\bar{\delta}$.
```

In line 1 we compute matrix F, containing the frequency values defined in (III.1). Summing a matrix over axis 1 means, in Numpy syntax, computing a column vector where each element is the sum of the corresponding matrix row – in our case, the total amount of calls to/from country c. Also notice that the division between a matrix and a column vector is permitted in Numpy: each element p_{ct} of P is thus divided by the corresponding value of the column – in other words, exactly the operation we do in (III.1).

Computing the average distances δ_{ct} defined in (III.2) is again done as a matrix operation, specifically, a row-by-column product between F and D (line 2).

Finally, the country-wise distances $\bar{\delta}_c$ are computed by multiplying element-wise Δ and F, and then computing the sum of each row of the resulting matrix. This is done in line 3: the $\cdot$ symbol indicates element-wise multiplication, and the sum over axis 1 has the same meaning as in line 1.

2) Map-reduce processing: Even with the vectorized implementation presented in Alg. 1, processing the whole trace at once is impossible: due to its sheer size, the trace cannot be loaded into memory. While it is true that there are ways to perform disk-based operations (most eminently, database indices), in-memory operations are typically faster and more flexible.

As mentioned in Sec. II, traces are distributed as compressed archives, containing one file per day. The size of these files averages 350 MByte, very convenient to load and process in-memory. Our approach is the following:

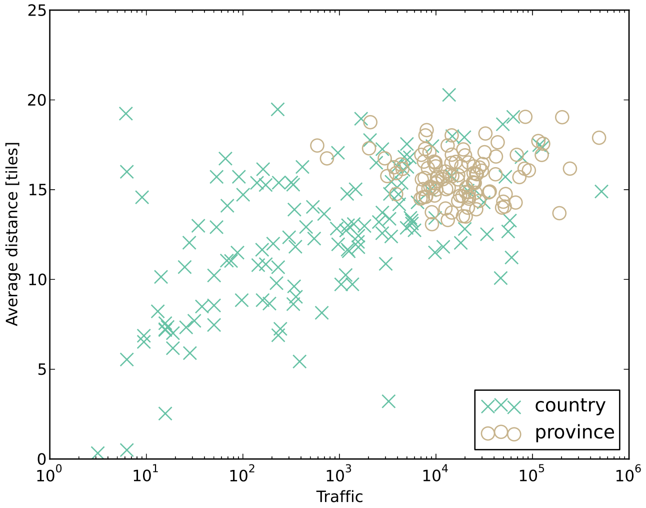


Fig. 1. Size and average distance for Italian provinces and foreign countries.

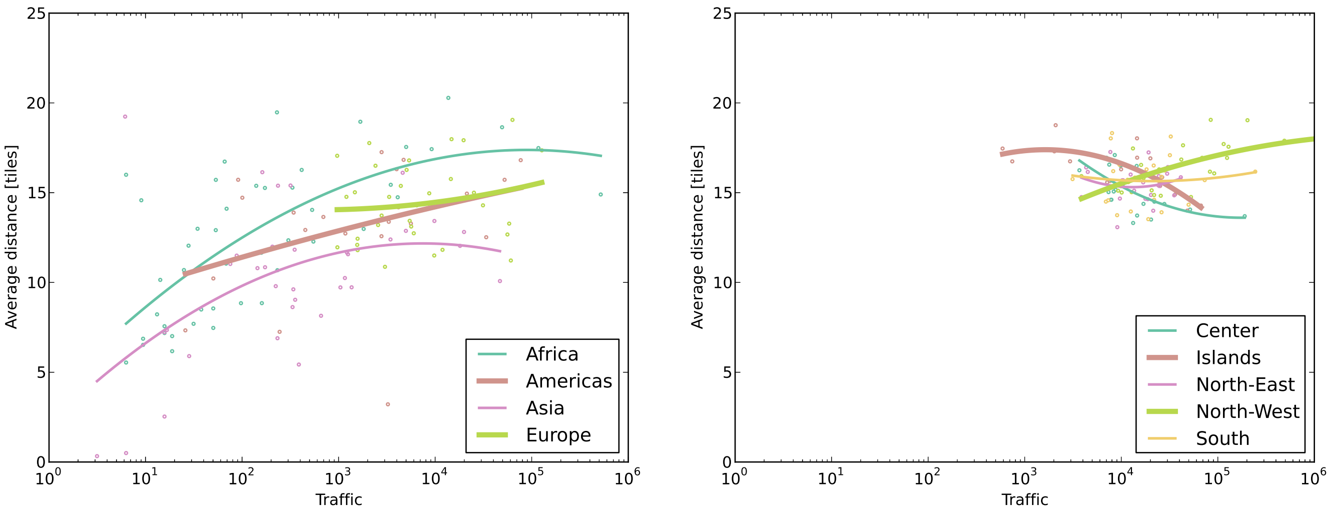
1) load each file into memory;
2) select and aggregate the relevant information in each file (*map* pass) and save it;
3) process the aggregate information of all files (*reduce* pass);
4) output the result.

Steps 2-3 above correspond to the *map* and *reduce* passes of the MapReduce programming model. Introduced by Google [9] and implemented by several open-source frameworks such as Apache Hadoop, MapReduce is one of the most popular and effective approaches to deal with "big data". The basic idea behind it is processing data one piece at the time, and then aggregating the resulting piece-wise outputs.

In our case, the input to the *map* pass is represented by individual trace files, whose format is shown in Tab. I and Tab. II. We remove the information we do not need, e.g., activities before 8pm, SMS and Internet. Then, we sum the volumes of incoming and outgoing calls, and we further sum over time intervals (the equivalent of a SQL `GROUP BY` operation). The output of the *map* step has the format summarized in Tab. III; its size is less than 1/100 of the input (mostly due to the fact that we discard time information).

In the *reduce* pass, we take the output of individual *map* passes, and again perform a `GROUP BY`-like operation, obtaining the p_{ct} values needed to compute our metric, as detailed in Sec. III-A.

Thanks to our MapReduce-like procedure, the size of our data never exceeds the one of our memory, making it possible to use the fast, in-memory tools described above. It is worth stressing, however, that we are *not* using MapReduce in a cluster, e.g., through Hadoop: all our processing took place on a single server – the total processing time was below 15 minutes, and the longest step was decompressing the



(a)

(b)

Fig. 2. Size and average distance for foreign communities (a) and Italian provinces (b), with polynomial fit for each continent and zone.

archives.

IV. Results

The first thing we are interested in is visualizing the relationship between the size of a community and the tendency of its people to cluster together.

In Fig. 1, each circle corresponds to an Italian province, and each cross to a foreign country. The coordinates of each point reflect the traffic generated by the corresponding community and the average distance between its members.

A first thing we can observe is that smaller communities tend, in general, to be more clustered – consistently with what one might expect. However, this is nothing like a general rule. Look, for example, at the communities whose traffic is sized at $2 \cdot 10^2$: their average distance ranges anywhere between 5 and 20 tiles.

It is also interesting to look at Italian provinces, represented by circles in Fig. 1. Their size and average distance are virtually equivalent to the ones of the biggest foreign communities. This suggests that the level of integration of such communities – or, at least, the tendency of their members to settle in various part of the city – is very high, equivalent to the one of Italian immigrants.

We have observed that the correlation between the size of a community and the distance between its members is quite loose. Now, we study how such a correlation changes for communities from different continents.

In Fig. 2(a), we highlight how the size/distance relationship changes for different continents. Asians seem to exhibit a stronger tendency to cluster together, while Africans are more evenly scattered throughout the city. Fig. 2(b) is built in the same way, but focuses on Italian provinces. Here, we see no clear trend: Italians from different parts of the country tend to cluster together to similar extents.

Tab. IV(a) summarizes the biggest communities. Romania and Senegal have the highest average distance, i.e., the weakest tendency to cluster together; Bangladesh has the shortest one. Also notice that, contrary to what one might expect, China does not show up among the biggest communities – perhaps the Chinese in Milan prefer landline phones or VoIP services.

Tab. IV(b) summarizes the communities with the shortest average distance. Most, but not all, such communities are very small. Exceptions include Haiti and Azerbaijan: their size is not exceptionally small, but their members live remarkably close to each other – recall that each tile is 235 meters in size.

TABLE IV
Foreign communities with highest traffic (a) and shortest average distance (b).

(a)

Nation	Total traffic	Average distance	Continent
Egypt	518550	14.89	Africa
Ukraine	127235	17.35	Europe
Senegal	117287	17.48	Africa
Ecuador	76894	16.81	Americas
Romania	63171	19.05	Europe
Peru	52347	15.71	Americas
Morocco	49291	18.64	Africa
Bangladesh	46858	10.07	Asia

(b)

Nation	Total traffic	Average distance	Continent
Cambodia	6	0.49	Asia
Uzbekistan	15	2.53	Asia
Haiti	3237	3.21	Americas
Azerbaijan	386	5.42	Asia
Niger	6	5.53	Africa
Vietnam	28	5.89	Asia
Zambia	18	6.17	Africa
Indonesia	232	6.89	Asia

Also notice how most of such ultra-clustered communities are Asian, consistently with the trend we see in Fig. 2(a).

V. Conclusions

We have exploited the communication traces shared by Telecom Italia with the participants to their Big Data Challenge to investigate the locality patterns of international calls in Milan.

We started by describing our metric of interest – the average distance between nationals of the same country, adding further details on how to efficiently compute it. Additionally, we discussed the MapReduce-like way in which we processed our traces.

Results highlighted that smaller communities have a slightly stronger tendency to cluster together. More importantly, such a tendency seems to depend on the continent of origin of each community, being strongest for Asians and weakest for Africans.

We ran the same analysis for Italian provinces, seeking for similarities and differences with foreign communities. Indeed, immigrants from Italian provinces are similar to the biggest foreign communities for numbers and tendency to cluster. However, there is no clear difference between immigrants from different parts of Italy, e.g., North and South.

References

[1] D. Bailey, C. Sodano, "Census: Maps show migration trends," BBC NEWS, http://www.bbc.co.uk/news/uk-20713380 [Accessed February 2014]

[2] Credit Suisse, "U.S. wireless networks running at 80% of capacity," http://benton.org/node/81874, 2011.

[3] B.S. Arnaud, "iPhone slowing down the Internet – Desperate need for 5G R&E networks," http://billstarnaud.blogspot.com/2010/04/iphone-slowing-down-Internet-desperate.html [Accessed May 2012].

[4] F. Malandrino, C. Casetti, C.-F. Chiasserini, Z. Limani, "Fast Resource Scheduling in HetNets with D2D Support," IEEE INFOCOM, Toronto, Canada, April 2014.

[5] K. Fitchard "Can LTE-broadcast dam the mobile video deluge?" http://gigaom.com/2013/01/10/can-lte-broadcast-dam-the-mobile-video-deluge/ [Accessed February 2014]

[6] M. Nati, A. Gluhak, F. Martelli, R. Verdone, "Measuring and Understanding Opportunistic Co-presence Patterns in Smart Office Spaces," IEEE GreenCom, August 2013.

[7] Telecom Italia Big Data Challenge 2014, http://www.telecomitalia.com/bigdatachallenge [Accessed February 2014]

[8] Numpy project, http://www.numpy.org [Accessed February 2014]

[9] D. F. Carr, "How Google Works," http://www.baselinemag.com/c/a/Infrastructure/How-Google-Works-1 [Accessed February 2014]