Inferring Pairwise Co-location from Noisy Bluetooth Signals

Constantin Teodor Gherghescu

Kongens Lyngby 2014
IMM-M.Sc.-2014-????

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Matematiktorvet, building 303B,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3351
compute@compute.dtu.dk
www.compute.dtu.dk IMM-M.Sc.-2014-????

Summary (English)

The goal of the thesis is to ...

Summary (Danish)

Målet for denne afhandling er at ...

Preface

This thesis was prepared at the department of Applied Mathematics and Computer Science at the Technical University of Denmark in fulfilment of the requirements for acquiring an M.Sc. in Computer Science and Engineering.

Lyngby, 20-June-2014

Constantin Teodor Gherghescu

Acknowledgements

I would like to thank my supervisors Sune Lehmann Jørgensen, Jakob Eg Larsen, and Vedran Sekara for their guidance and feed-back. Additionally, I would like to thank Arek Stopczynski for his help with the database, and Piotr Sapieżyński and Ole Winther for their help with inferring methods.

Contents

Chapter 1

Introduction

1.1 Motivation

Epidemiology [1], personal health issues [2], group discovery [3], human mobility [4, 5], efficient team creation [6, ?], the analysis of academic success [7], network theory [8], also Fig. 1.1, and psychological research [9]; all of the above have begun making use of the same notion, one that is difficult to quantify [10, 11]: social connections and social interactions between individuals. And although co-location does not necessarily mean physical social interaction, it is a requirement for it. [12].

There are a number of approaches to determining co-location: self-reported data, a more traditional approach, which is prone to cognitive bias, social desirability bias and halo error [14, 15]. A more recent approach involves the use of data provided by modern means of communication, namely mobile phones. This data can come from both the actual phone, in the form of GPS locations or bluetooth readings [13], as well as from the phone company itself in the form of anonymous cell tower records [16, 17]. This has the advantage of being applicable almost anywhere, because of the high percentage of mobile phone penetration (95.5% estimated by the International Telecommunication Union in May 2014). However, there are approaches that yield better results, but come with financial, environmental or other types of additional cost: RFID tags [18],

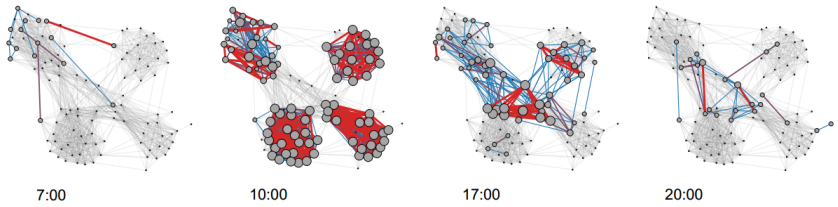

Figure 1.1: Face-to-face interactions over a day for college students. Blue means a low, and red a high frequency of interactions. Image from [13]

.

audio-video recordings [19], on-body sensors [20] and wifi signals [21], just to name a few.

1.2 Objective

Given a bluetooth RSSI between two phones, the purpose of this paper is to indicate a method which reliably and accurately determines the existence of pairwise co-location between the people carrying the two phones.

Reliability refers to the fact that multiple tries with the same input, and constant settings (same machine learning algorithm, same parameters and same training data), will always yield exactly the same result. While accuracy refers to the precision of the algorithm during cross-validation testing, or how close to 100% it is.

This inference is achieved by applying and analysing multiple machine learning algorithms on a data set consisting of two main parts:

- Data automatically recorded by the SensibleDTU data collector app [13]
- Ground truth data obtained by the test subjects by interacting with the FriendFinder app

For each machine learning algorithm the parameter configuration which yields the best results will be chosen, followed by a comparison between the best configurations for each algorithm.

1.3 Scope and limitations

The paper analyses the data obtained from three test subjects. Each subject has been given the same phone model, Samsung Galaxy Nexus. The app built for the phones, FriendFinder, has a purely functional purpose. As such, little to no consideration has been given to the style, theme or design of the app. Fig 2.1 shows the app, and while fir for purpose, it is not very aesthetic.

While considerable testing has been done with regards to the algorithm parameters, the machine learning algorithms list is by no means exhaustive. There are three main algorithms: Naive Bayes, Neural Networks, and Recursive learning, and an explanation on why a fourth, Hidden Markov Model, is unsuitable for this type of data.

When analysing data, we only look at the bluetooth RSSI value, and data derived directly while measuring it. For example, given that measurements are taken every five minutes, we at some point look at the length of an uninterrupted string of measurements, or at the measurements taken before and after (if possible) a specific measurement. GPS traces, phone records, infrared sensors, or facebook/email information are not taken into consideration.

1.4 Thesis Outline

The thesis begins with this introduction, which gives an overview of the general theme of the project. The objectives, scope and limitations and thesis outline are all self-defining.

It continues with a section which describes the data acquisition process and the methodology used to obtain the data from the SensibleDTU database. The section also describes the FriendFinder app, as well as its implementation.

Next, the machine learning algorithms are presented. For each algorithm a theoretical overview is given. The implementation details are discussed, followed by the results obtained by applying it to the data. At the end of this chapter, we do a comparative analysis between the algorithms, followed by the last section, the conclusions, where we will give the final results and recommendations.

Chapter 2

Data acquisition

The data used in the paper has been gathered over the course of three months, starting in early March and ending in late May. The data has been obtained with the help of three student volunteers. Each volunteer carried a phone, provided by the SensibleDTU project, the same project this thesis is a part of [22, 13]. The phones are Samsung Galaxy Nexus [23] and are running the Android operating system [24].

Each phone came with two apps. The first recorded regularly a multitude of information, such as bluetooth RSSI, GPS traces, battery usage and cell tower information, and is a part of the SensibleDTU project [13], while the second allowed the volunteers to manually name the person they are in physical proximity with, and was developed during this thesis. The second app provided the ground-truth information.

This gave rise to two types of data. First, a continuous stream of regular measurements from the first app, and punctual messages from the second app. Below we will go into more detail about how exactly the two types of data look, how are they gathered and finally, how the two are combined to create a unitary set which serves as a basis for the machine learning algorithms.

2.1 FriendFinder app

2.1.1 App overview and implementation

As the app, FriendFinder, is made for the Android operating system, it is implemented using Java for Android [25]. It has a Google App Engine backend [26]. Fig 2.1 shows the main screen of the app.

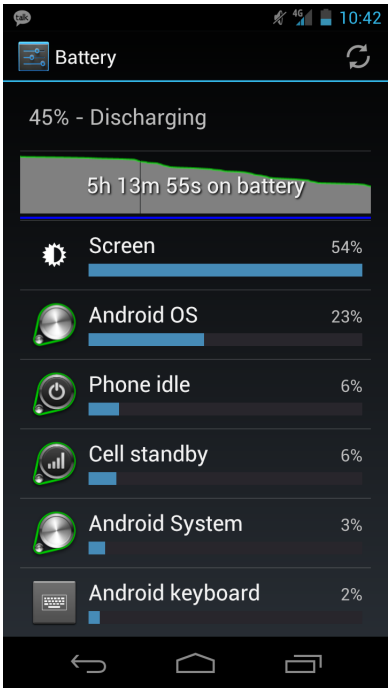

Figure 2.1: FriendFinder app

.

FriendFinder is implemented using a client server architecture, where the clients are the apps installed on the phone, and the server is the Google App Engine. Fig 2.2 shows an overview of this particular architecture. In this case, the FriendFinder app (in the role of the client) makes a *save data* request to the Google App Engine (the server). The server in turn responds with the result of the operation, either a *succes* or *failure* message.

The app contains a single screen (the one showed in Fig 2.1), which corresponds to the main activity. Activities are the main building blocks of an Android app,

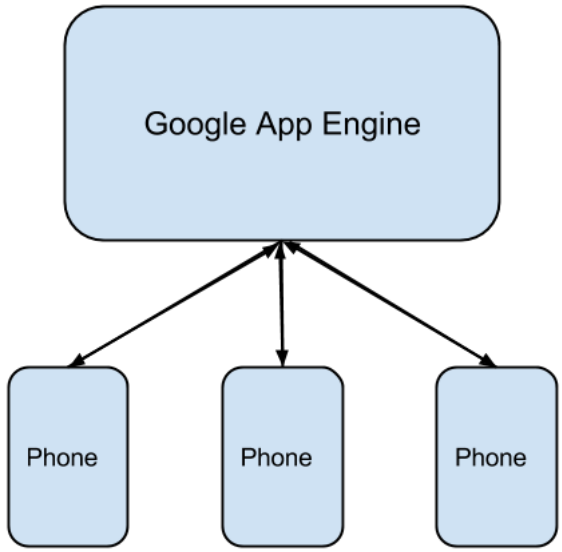

Figure 2.2: Client Server Architecture used by the FriendFinder app

.

and they are used both to interact with the user, as well as provide additional functionalities [27]. On the main screen there are buttons for each volunteer. Once one of the volunteers is in physical proximity with another volunteer (as perceived by either one), they both press the button corresponding to each other. A confirmation message is displayed, as can be seen in Fig. 2.3.

For the data to reach the database on the Google App Engine, an internet connection is required. However, the app can also function off-line, by saving all the data locally, and sending it to the on-line database as soon as an internet connection is established. It does this by using an IntentService [28], which has two main functions:

- A first function is to store the data locally, until an internet connection is established.
- the second, and most important, is to check periodically for an internet connections. Once an internet connection has been established, all the data is sent to the Google App Engine. The app checks every one minute for internet access. This allows for a relatively fast updating of the database, while at the same time keeping the resource use at a level

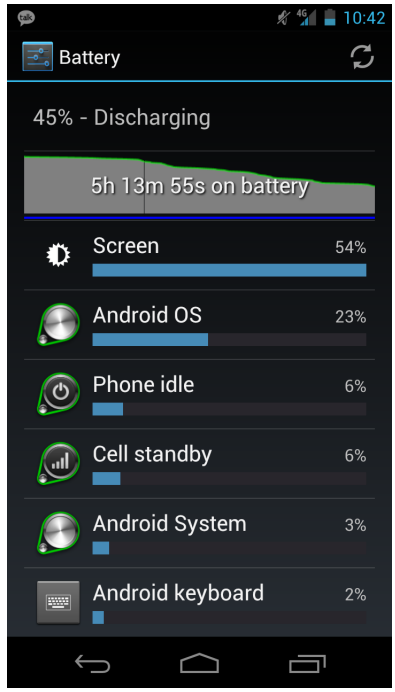

Figure 2.3: Confirmation message on the FriendFinder app

.

that does not impede the normal functioning of the phone.

The data is stored locally in RAM of the phone (the volatile memory). This is made possible by the relatively small size of the data being saved (the names of the two people involved, and an ID object that also serves as a timestamp) and the memory capacity of the phone (approximately 700 MB of RAM). However, this has the disadvantage of being vulnerable to the phone turning off due to lack of battery. The volunteers have been made aware of this fact.

Once the internet connection is established, the actual sending of the data is a simple matter, done with the help of the Google App Engine API. One thing to mention here is that each data entry is sent individually. If at any point, a *save data* request is met by a *failure* answer, the request is repeated until the *success* message is received. In case of *failure*, subsequent attempts have a one second delay between them. This is done to ensure that the app does not impede the overall functionality of the phone.

2.1.2 Data

Once a button with the name of a person is pressed, the data that is saved on the phone, and eventually sent and saved on the online database has the following format:

```
{ID, owner, target}
```

ID It has a double role. First, the ID is a unique identifier, used for differentiating between multiple entries, as well as for various database operations. This field is required by the Google App Engine. Secondly, the ID also plays the role of the timestamp, used in future computations. While there are valid concerns that the ID might not be unique, due to two people pressing the button at the same time, the fact that the timestamp also includes the millisecond, and the fact that there are only three participants in the project results in an extremely low probability of identical timestamps. Obviously, the timestamp corresponds to the moment the button was pressed, and not the moment the data was sent to the database.

owner This field refers to the owner of the phone, which indicates the person that has made the observation.

target This field names the person that has been observed by the owner.

Fig. 2.4 shows an example of how the data looks, with the caveat that the *friend* field in the image corresponds to the *target* field in the description above. The saved data was accessed and retrieved by API calls to the Google App Engine.

2.2 SensibleDTU data

‹ Prev 20 **21-40** Next 20 ›

ID/Name	friend	owner
name=2014-04-02 11:24:37.318	rafa	Magda
name=2014-04-02 11:24:58.803	ferdinando	Rafa
name=2014-04-02 12:52:05.905	ferdinando	Rafa
name=2014-04-02 12:52:08.734	rafa	Ferdinando
name=2014-04-06 10:35:34.509	magda	Rafa
name=2014-04-06 10:37:02.683	magda	Ferdinando
name=2014-04-06 10:37:04.752	rafa	Ferdinando
name=2014-04-06 10:37:37.694	ferdinando	Rafa
name=2014-04-06 10:44:09.723	bartosz	Rafa
name=2014-04-06 13:01:40.166	ferdinando	Magda
name=2014-04-06 13:01:42.319	rafa	Magda
name=2014-04-07 20:50:01.430	rafa	Magda
name=2014-04-07 20:50:06.456	magda	Rafa
name=2014-04-09 09:56:46.845	bartosz	Rafa

Figure 2.4: Saved entries for the FriendFinder app

Chapter 3

Methods for inferring pairwise co-location

3.1 Naive Bayes

3.1.1 Original

3.1.2 V1

3.1.3 V2

3.2 HMM

3.3 Recursive data

3.4 Neural Networks

Appendix A

Stuff

This appendix is full of stuff ...

Bibliography

[1] L. A. N. A. H. E. S. Y. b. Fredrik Liljeros, Christofer R. Edling, "The web of human sexual contacts," 2001.

[2] A. Madan, S. T. Moturu, D. Lazer, and A. S. Pentland, "Social sensing: Obesity, unhealthy eating and exercise in face-to-face networks," in *Wireless Health 2010*, WH '10, (New York, NY, USA), pp. 104–110, ACM, 2010.

[3] S. Mardenfeld, D. Boston, S. Pan, Q. Jones, A. Iamntichi, and C. Borcea, "Gdc: Group discovery using co-location traces," in *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pp. 641–648, Aug 2010.

[4] J. Sun, J. Yuan, Y. Wang, H. Si, and X. Shan, "Exploring space–time structure of human mobility in urban space," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 5, pp. 929 – 942, 2011.

[5] A. Sevtsuk and C. Ratti, "Does urban mobility have a daily routine? learning from the aggregate data of mobile networks," *Journal of Urban Technology*, vol. 17, no. 1, pp. 41–60, 2010.

[6] O. Bandiera, I. Barankay, and I. Rasul, "Social connections and incentives in the workplace: Evidence from personnel data," *Econometrica*, vol. 77, no. 4, pp. 1047–1094, 2009.

[7] D. Blansky, C. Kavanaugh, C. Boothroyd, B. Benson, J. Gallagher, J. Endress, and H. Sayama, "Spread of academic success in a high school social network," *PLoS ONE*, vol. 8, p. e55944, 02 2013.

[8] Y. ong-Y. eol Ahn, J. ames P. . Bagrow, and S. une Lehmann, "Link communities reveal multiscale complexity in networks," vol. - 466, no. - 7307, pp. – – 764, - 2010/08/05/print.

[9] K. K. Rachuri, M. Musolesi, C. Mascolo, P. J. Rentfrow, C. Longworth, and A. Aucinas, "Emotionsense: A mobile phones based adaptive platform for experimental social psychology research," in *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, Ubicomp '10, (New York, NY, USA), pp. 281–290, ACM, 2010.

[10] G. ergely Palla, A. lbert-L. aszlo Barabasi, and T. amas Vicsek, "- Quantifying social group evolution," vol. - 446, no. - 7136, pp. – – 667, - 2007/04/05/print.

[11] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao, "User interactions in social networks and their implications," in *Proceedings of the 4th ACM European Conference on Computer Systems*, EuroSys '09, (New York, NY, USA), pp. 205–218, ACM, 2009.

[12] N. Eagle, A. S. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *Proceedings of the National Academy of Sciences*, vol. 106, no. 36, pp. 15274–15278, 2009.

[13] A. Stopczynski, V. Sekara, P. Sapiezynski, A. Cuttone, J. E. Larsen, and S. Lehmann, "Measuring large-scale social networks with high resolution. working paper," *CoRR*, vol. abs/1401.7233, 2014.

[14] S. Wuchty, "What is a social tie?," *Proceedings of the National Academy of Sciences*, vol. 106, no. 36, pp. 15099–15100, 2009.

[15] R. M. Gonyea, "Self-reported data in institutional research: Review and recommendations," *New Directions for Institutional Research*, vol. 2005, no. 127, pp. 73–89, 2005.

[16] J.-P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, K. Kaski, J. Kertész, and A.-L. Barabási, "Structure and tie strengths in mobile communication networks," *Proceedings of the National Academy of Sciences*, vol. 104, no. 18, pp. 7332–7336, 2007.

[17] P. Hoevel and A.-L. Barabasi, "Temporal and spatial regularity of mobile-phone data," in *APS Meeting Abstracts*, p. 54005, Feb. 2012.

[18] A. Barrat and C. Cattuto, "Temporal networks of face-to-face human interactions," in *Temporal Networks* (P. Holme and J. Saramäki, eds.), Understanding Complex Systems, pp. 191–216, Springer Berlin Heidelberg, 2013.

[19] D. Wyatt, T. Choudhury, and H. Kautz, "Capturing spontaneous conversation and social dynamics: A privacy-sensitive data collection effort," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4, pp. IV–213–IV–216, April 2007.

[20] D. Olguin, B. Waber, T. Kim, A. Mohan, K. Ara, and A. Pentland, "Sensible organizations: Technology and methodology for automatically measuring organizational behavior," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, pp. 43–55, Feb 2009.

[21] M. B. Kjærgaard and P. Nurmi, "Challenges for social sensing using wifi signals," in *Proceedings of the 1st ACM Workshop on Mobile Systems for Computational Social Science*, MCSS '12, (New York, NY, USA), pp. 17–21, ACM, 2012.

[22] "Sensibledtu." https://www.sensible.dtu.dk/, June 2014.

[23] "Samsung galaxy nexus." http://www.samsung.com/us/mobile/cell-phones/GT-I9250TSGGEN, June 2014.

[24] "Android." http://www.android.com/, June 2014.

[25] "developer.android.com." http://developer.android.com/, June 2014.

[26] "Google app engine." https://developers.google.com/appengine/, June 2014.

[27] "Android activity." http://developer.android.com/reference/android/app/Activity.html, June 2014.

[28] "Intentservice." http://developer.android.com/reference/android/app/IntentService.html, June 2014.