Understanding email communication patterns

Daniel Smilkov
M.Sc. Computer Science, 2011
B.Sc. Computer Science, 2009
Ss. Cyril and Methodius University, Skopje, Macedonia

Submitted to the
Program in Media Arts and Sciences,
School of Architecture and Planning,

in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Media Technology
at the **Massachusetts Institute of Technology**

June 2014

Signature redacted

Author ______________________________

Daniel Smilkov
Program in Media Arts and Sciences
May 16, 2014

Signature redacted

Certified By ______________________________

Dr. César A. Hidalgo
Assistant Professor in Media Arts and Sciences
Program in Media Arts and Sciences

Signature redacted

Accepted By ______________________________

Dr. Pattie Maes
Professor of Media Technology, Associate Academic Head
Program in Media Arts and Sciences

Understanding email communication patterns

Daniel Smilkov

Submitted to the

Program in Media Arts and Sciences,
School of Architecture and Planning,

On May 16, 2014

in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Media Technology
at the **Massachusetts Institute of Technology**

Abstract

It has been almost two decades since the beginning of the web. This means that the web is no longer just a technology of the present, but also, a record of our past. Email, one of the original forms of social media, is even older than the web and contains a detailed description of our personal and professional history. This thesis explores the world of email communication by introducing Immersion, a tool build for the purposes to analyze and visualize the information hidden behind the digital traces of email activity, to help us reflect on our actions, learn something new, quantify it, and hopefully make us react and change our behavior. In closing, I look over the email overload problem and work-life balance trends by quantifying general email usage using a large real-world email dataset.

Advisor

Dr. César A. Hidalgo
Assistant Professor in Media Arts and Sciences
Program in Media Arts and Sciences

Understanding email communication patterns

Daniel Smilkov

Submitted to the

Program in Media Arts and Sciences,
School of Architecture and Planning,

On May 16, 2014

in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Media Technology
at the **Massachusetts Institute of Technology**

June 2014

Signature redacted

Advisor ______________________________

Dr. César A. Hidalgo
Asahi Broadcast Corporation Career Development Professor
Massachusetts Institute of Technology

Signature redacted

Reader ______________________________

Dr. Sepandar Kamvar
LG Career Development Professor of Media Arts and Sciences
Massachusetts Institute of Technology

Signature redacted

Reader ______________________________

Dr. Alex 'Sandy' Pentland
Toshiba Professor of Media Arts and Sciences
Massachusetts Institute of Technology

ACKNOWLEDGMENTS

I am truly grateful to so many people that it is impossible to acknowledge all of them, and I hope everyone that has helped me knows how extremely thankful I am to have you in my life.

I would like to express a deep appreciation to my advisor **César Hidalgo** for giving me the opportunity to come to MIT, my dream school. Thank you for continually conveying a spirit of adventure and excitement in regard to research and helping me broaden my horizon. Without your guidance and countless hours of persistent help, and your extreme availability, this thesis would have not been possible. I would also like to thank my readers **Sep Kamvar** and **Alex Pentland** for their valuable feedback. I'm very grateful to have you both as readers. I am and will always be grateful to my lifelong mentor and role model **Ljupco Kocarev**. I hope to collaborate with you again in the future.

I would like to thank my research group **Macro Connections.** Thank you guys for being great teammates, officemates and friends. A special thank you to **Deepak Jagdish**, my equal partner in the project behind this thesis. Without you, Immersion would have never come to fruition.

Thanks also to my best friend **Alexandre Sahyoun**. I was extremely lucky to have you as a roommate for the last 2 years. We helped each other, and pushed each other, in perfect balance. Thanks to my friends at MIT, and outside, for all the weekends and fun times we shared together. We worked hard and played hard. I will miss you all.

Finally, I would like to thank my parents, **Tihomir Smilkov** and **Suzana Smilkova,** and my sister **Milena Angova,** for supporting and guiding me through every decision I've made in life.

TABLE OF CONTENTS

1. INTRODUCTION

Thanks to the Internet, long-distance communication is essentially free. To communicate with someone, we don't have to travel around the world, or even to the next room. The only cost in email, messaging on social networks or sending a text message, is the time it takes to read and write the message. As a result, online communication is becoming more and more important in our everyday life. As we exchange messages using modern platforms, we leave an immense amount of digital traces containing a detailed description of our personal and professional history. The goal of this thesis is to analyze and visualize the information hidden behind these digital traces, to make the invisible visible, and ultimately, to make us react and change our behavior when exposed to this information. In particular, this thesis explores the world of email communication, one of the original forms of social media, by introducing Immersion, an online platform that provides interactive data visualization of one's personal and professional social network. I created Immersion together with my colleague Deepak Jagdish and my advisor Cesar Hidalgo, which was released in July 2013, and at the time of writing this thesis it is still evolving, both as a visualization tool, and as a framework for future analysis of email data.

This thesis is organized as follows. This first chapter provides the motivation behind Immersion, a brief history of network visualization, which one of the main components in Immersion, and related works. Chapter 2 and 3 focus on Immersion itself, which was developed in two major sprints, resulting in two versions of the software, each explained in detail in each chapter. Chapter 4 provides initial exploratory data analysis of global email trends by analyzing a longitudinal email dataset comprising of 180,000 people and 4.7 billion email headers. Lastly, Chapter 5 concludes the thesis by talking about future work and the lessons I learned from making Immersion.

The motivation behind email

Nowadays, new communication technologies are being developed every year. Today's Internet is flooded with social networking platforms such as Facebook, Twitter, Google chat and new phone and desktop messaging apps, replacing old technologies such as ICQ and MSN Messenger. Email, however, hasn't been replaced by these new technologies even though its technology is more than 20 years old. It co-exists with Facebook and Twitter because it provides a different form of online communication. With its long history, email provides an extensive record of our past and this historical property of email was one of our motivations to play with email data.

Another motivation was the sheer size of email. Everyone knows that email is big, but what is more surprising, is that it is still growing. In fact, I had many debates concerning the stagnating usage of emails, and some people even argued that email usage is declining. To provide a better perspective, and an argument for those that think otherwise, I will share two numbers concerning the change in email usage from 2012 to 2013. A recent study (The Radicati Group, 2012; The Radicati Group, 2013) shows that the number of emails exchanged in 2013 is 26% more than 2012 and that email gained 252 million new unique users, which is roughly 80% of the population of the USA. Email is the fastest growing communication technology, faster than any social network or phone app.

The third motivation for diving into the world of email was the intrinsically personal nature of email data. There are no options to share, like or pin your email to a public wall. Email users expect total privacy when it comes to their data. However, with the revelation that the National Security Agency has been collecting data about millions of Americans' phone calls, emails and credit card transitions, the "guaranteed" privacy of every individual was brought to question. Furthermore, a large body of research literature points to some fundamental limits of privacy, showing that true data anonymization is very hard to achieve (Sweeney, 2002; Bayardo & Agrawal, 2005; de Montjoye, Hidalgo, Verleysen, & Blondel, 2013). Following this trend, new platforms like openPDS have been developed recently (de Montjoye, Wang, & Pentland, 2012), that allow storing of data on the cloud and sharing it with third parties in a way that protects user's privacy by providing anonymous answers instead of raw data. With this strong trend of personal data sensing, storage and ownership, we realized that email was the perfect example to educate the general public about data privacy. The type of data collected by the government is nevertheless limited in scope, and contains only parts of the information behind an online transaction. This type of data is often referred to as *metadata*. In the case of email, metadata includes information about the sender, the recipients and the time of the email, but not the actual content. Thus, one of our goals was to demonstrate what email metadata, which the government agencies have been collecting for years in total secrecy, could reveal about a person.

Lastly, email is corporate. When we made Immersion, I didn't realize how important this property was. After the launch, we were contacted by tens of companies and organizations expressing how much they liked the tool and giving us feedback for new features. We learned that the analysis of email data could have an immense value for a company or an organization. Email data can reveal the implicit social network induced by internal communication, which can be significantly different from the hierarchical structure imposed by company's policies. Furthermore, algorithms can signal about employees that have significantly changed their way of communication and/or are about to switch companies. Email data may be used to predict and prevent potential frauds and leaking of company's information as well as to measure the performance of a company and its employees, and the effect of new internal policies. For example, when a company moves its employees from a closed-office space to an open-office space, one way of measuring the impact it has on productivity, is to compare past and recent email communication data (also known as A/B testing). Finally, I would like to stress that while Immersion works only with data at the individual level, and it's features are still far from the organizational applications I just described, the design principles, algorithms and lessons learned from building Immersion can provide a solid starting point towards solving organizational problems.

In the end, I would like to remind the reader that email data is just one specific part of data, and, as such, is inherently limited. The true potential behind data analysis lies in the symbiosis of signals of different nature, ranging from emails, phone calls, geospatial and proximity data, to heart rate and blood pressure time series (Eagle & Pentland, 2006; World Economic Forum, 2011; Pentland, 2009). There is no doubt that email data can provide us with interesting insights, but we should be mindful of its limitations and careful when deriving conclusions from it.

Brief history of network visualization

Since the main component in Immersion is the visualization of the social network, I believe I owe the reader of this thesis a brief history of network visualization.

Social networks are growing fast in popularity, and with that it becomes increasingly important to understand and analyze their properties and dynamics efficiently. However, as networks grow in size and complexity, simple statistics and mathematical tools are not able to answer all of our questions. Nowadays, visual metaphors are becoming a strong supplement to the classical statistical methods used to understand network dynamics (Correa & Ma, 2011).

The concept of visualizing networks is not new. The most common type of graphic representation of a network is the sociogram (Moreno, 1946), introduced by Moreno about 80 years ago. Moreno's motivation was to analyze choices and preferences within a group of people. The two major elements in the network are the actors, represented as circles, and links, which are represented as lines between these elements. The links can be drawn on the basis of many different criteria: social relations, channels of influence, preferences, lines of communications etc. A sociogram of 34 people in a karate club and their friendships is shown in Figure 1.

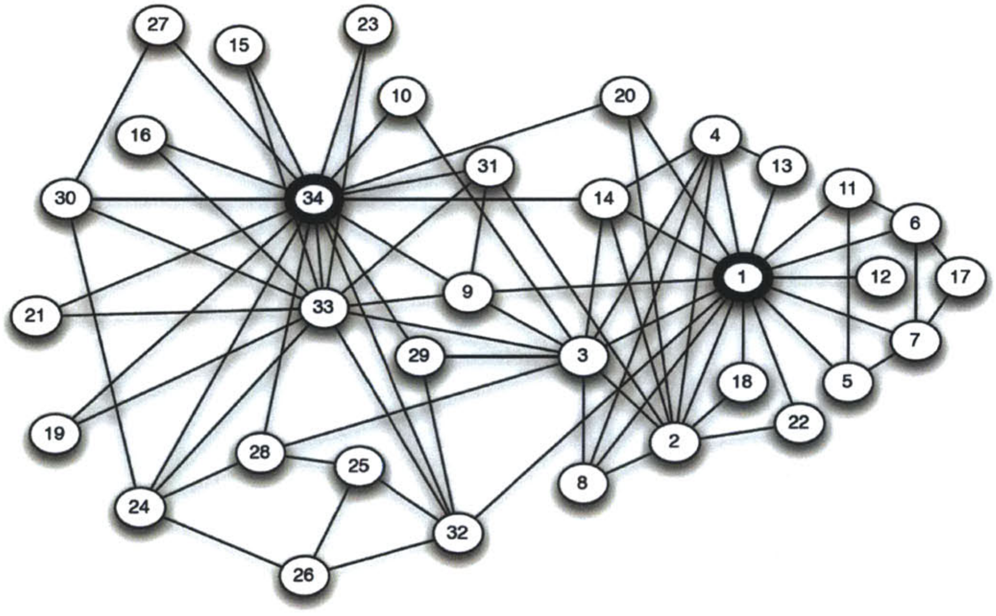


Figure 1 The social network of friendships within a karate club as depicted in the book of (Easley & Kleinberg, 2010).

While the concept of the sociogram is fairly old, due to lack of drawing software, most of the network visualizations were limited to a small number of nodes. In recent years, with the explosion of graph drawing tools like Pajek (Batagelj & Mrvar, 1998), JUNG (O'Madadhain, 2003) and Gephi (Bastian, Heymann, & Jacomy, 2009), among others, the number of networks visualizations on the Internet has increased exponentially.

Many of these software tools were designed for general graph analysis, and include standard network metrics, such as degree distributions, assortativity, clustering etc., as well as other types of visualizations such as histograms and scatterplots. These software tools still remain popular, but today's technology and the ability of computers to quickly filter out and slice large datasets, brought a new era of visualizations. The intersection between data exploration and human-computer interaction, made it possible for a new set of graphical information representation with dynamic reactive visualizations. This trend of interactive data visualizations has opened new possibilities for data exploration (Ward, Grinstein, & Keim, 2010). Users of online social networking sites can explore their social networks using third-party interactive visualizations apps like TouchGraph for Facebook (TouchGraph), InMaps for LinkedIn (LinkedIn), and MentionMapp for Twitter (MentionMapp). A similar tool for email data was missing however, which strengthened our motivation to create Immersion.

The era of personal analytics

We are also entering an era of personal analytics (Wolfram, 2012). As we spent more and more time interacting with technology, we leave a myriad of signals about our actions and behaviors. Our realization of the potential of personal data started a paradigm shift resulting in various communities, such as the Quantified Self movement (Quantified Self, Wikipedia). A new wave of personal data visualization engines sparkled a plethora of new companies, apps and tools, out of which I will point to the Wolfram Alpha personal analytics for Facebook (Wolfram Alpha). As these tools helped us realize the power of these digital personal signals, we entered into one of the biggest battles of the Internet - taking control of our own data (Schneier, 2013; World Economic Forum, 2011; de Montjoye, Wang, & Pentland, 2012). The idea of privacy and ownership of data had a strong play in the design of Immersion, and is prominently discussed on the Immersion website.

Related work

In this section I briefly review prior work closely related to email visualization and analytics. The email mountain project (Viegas F. , 2005), shown in Figure 2, provides a mountain view of a person's email archive with each layer of the mountain representing a different person. Another project from the same author is Themail (Viégas, Golder, & Donath, 2006), a word-based exploration of the email archive, where the primary design is organized into columns with each column of words referring to emails exchanged in a particular month with the selected person (see Figure 3). The main motivation behind Themail is similar to that of Immersion. In Viégas words:

> "Most tools for handling email archives have focused on the task of finding either specific messages or the 'important' emails. Less attention has been paid to the overall patterns of communication that can be gleaned from the daily accumulation of messages over time. Nevertheless, the patterns of communication we build up over time are themselves significant. As email archives grow, they become valuable records of people's relationships."

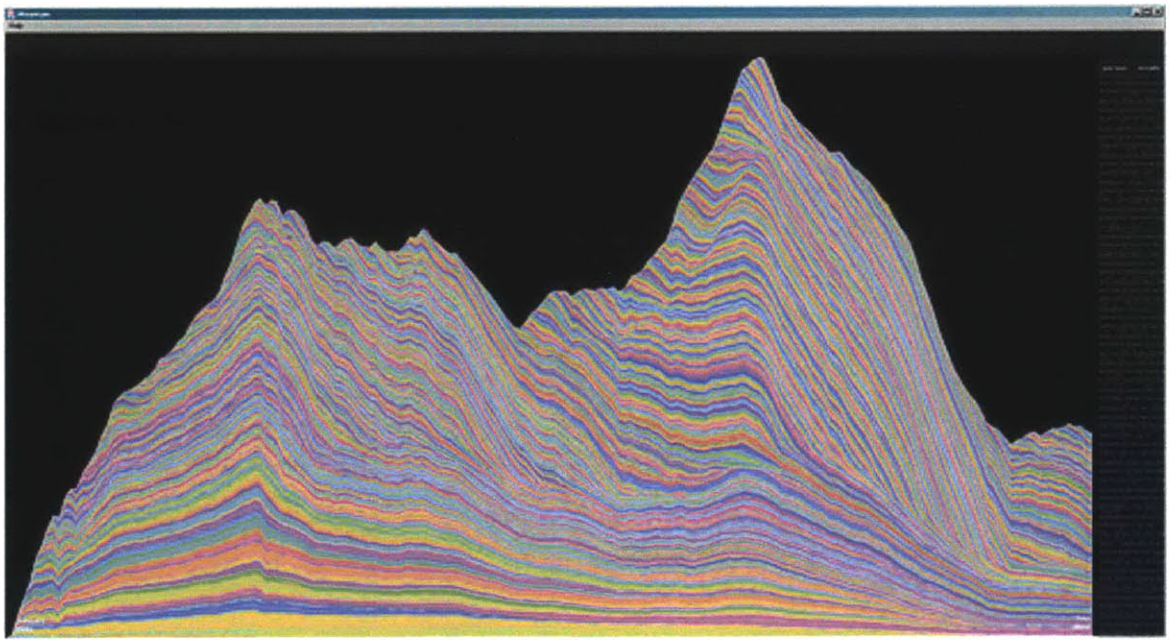


Figure 2 The Mountain email project.

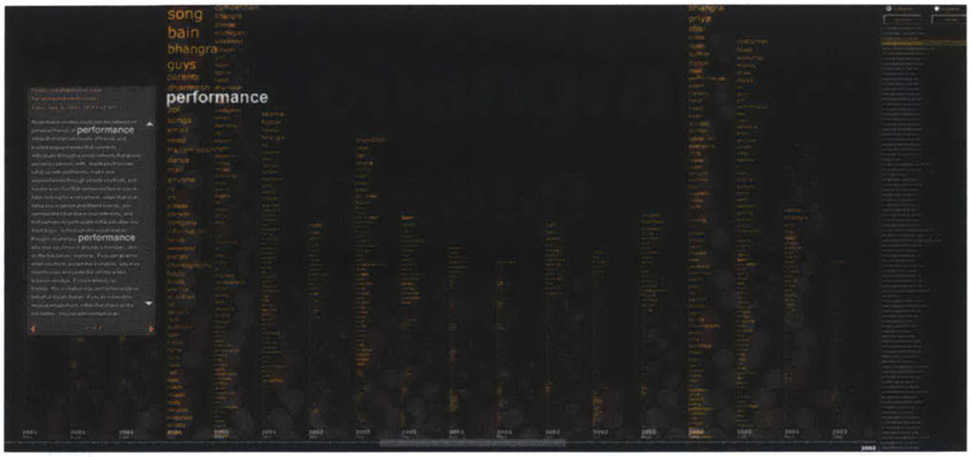
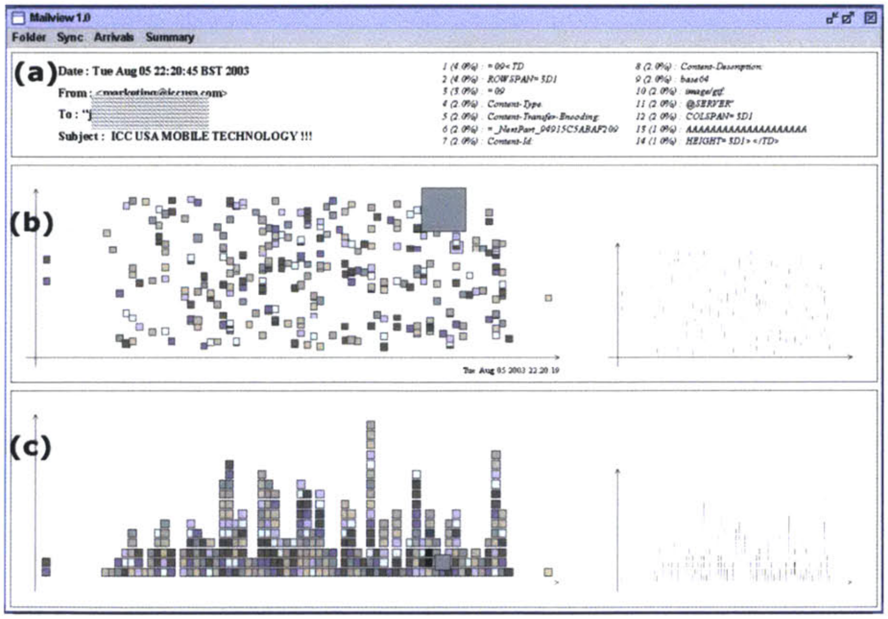


Figure 3 Themail, a visualization that portrays relationships using the interaction histories preserved in email archives.

Another related project is the work of (Perer & Smith, 2006), which aims to quickly capture contrasting portraits of email practices by providing different visualizations that contain hierarchical, correlational, and temporal patterns present in user's email repositories. Finally, an earlier experiment of interactive email visualization is Mailview (Frau, Roberts, & Boukhelifa, 2005), which utilizes filter and coordination techniques to explore archived email data. The emails are displayed on time-dependent plots enabling users to observe trends over time and identify emails with similar features. The interface of Mailview is shown on Figure 4.



Date : Tue Aug 05 22:20:45 BST 2003
From : [redacted]
To :
Subject : ICC USA MOBILE TECHNOLOGY !!!

1 (4.0%) : =09<TD
2 (4.0%) : ROWSPAN=3D1
3 (3.0%) : =09
4 (2.0%) : Content-Type
5 (2.0%) : Content-Transfer-Encoding
6 (2.0%) : =_NextPart_94915C5ABAF209
7 (2.0%) : Content-Id
8 (2.0%) : Content-Description
9 (2.0%) : base64
10 (2.0%) : image/gif
11 (2.0%) : @SERVER"
12 (2.0%) : COLSPAN=3D1
13 (1.0%) : AAAAAAAAAAAAAAAAAAA
14 (1.0%) : HEIGHT=3D1></TD>

Figure 4 The interface of Mailview.

2. IMMERSION

It has been almost two decades since the beginning of the web. This means that the web is no longer just a technology of the present, but also, a record of our past. Email, one of the original forms of social media, is even older than the web and contains a detailed description of our personal and professional history. Motivated by this observation in July 2013, we launched Immersion (Smilkov, Jagdish, & Hidalgo, 2013), a web tool that combines, analyzes and visualizes email data. Immersion allows Gmail, Yahoo and MS Exchange users to dive into the history of their email life by visualizing how their ego networks evolved over time. When we created Immersion, our philosophy was not to center email around timestamps or messages. This required us to adopt a representation in which people, and social links are the essential component.

This chapter talks about the first version of Immersion, which we launched in July 2013. While writing this thesis, we are in the process of finishing the new version of Immersion, which includes email subjects and natural language analysis, thus greatly expanding the capabilities of the tool. I will talk about the second version of Immersion in the next chapter.

Design

Let me quickly explain the interface of Immersion. Upon logging in and downloading the email data, Immersion shows the contact network that is automatically inferred from the email headers of the user (see Figure 5) where each person is represented as a circle and the size of the circle corresponds to the communication intensity, measured as a function of the number of exchanged emails with that person. The line between two circles denotes that those people were in an email conversation together. Below each circle, Immersion shows the name of the person.

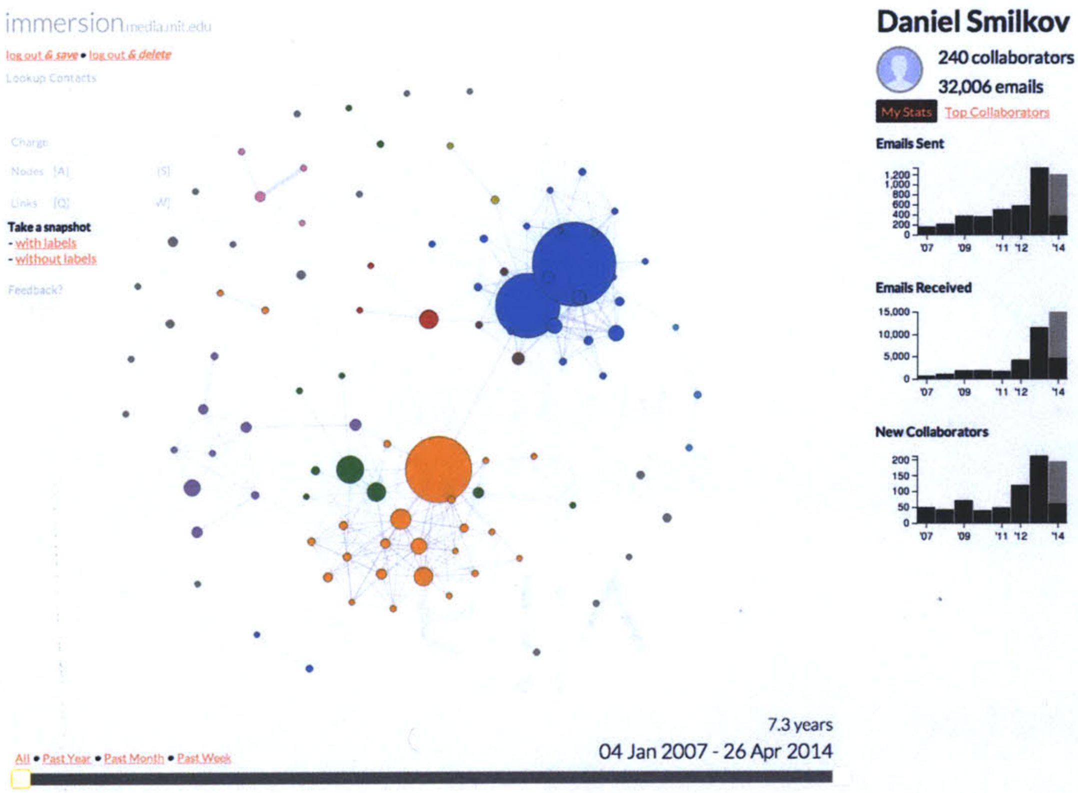
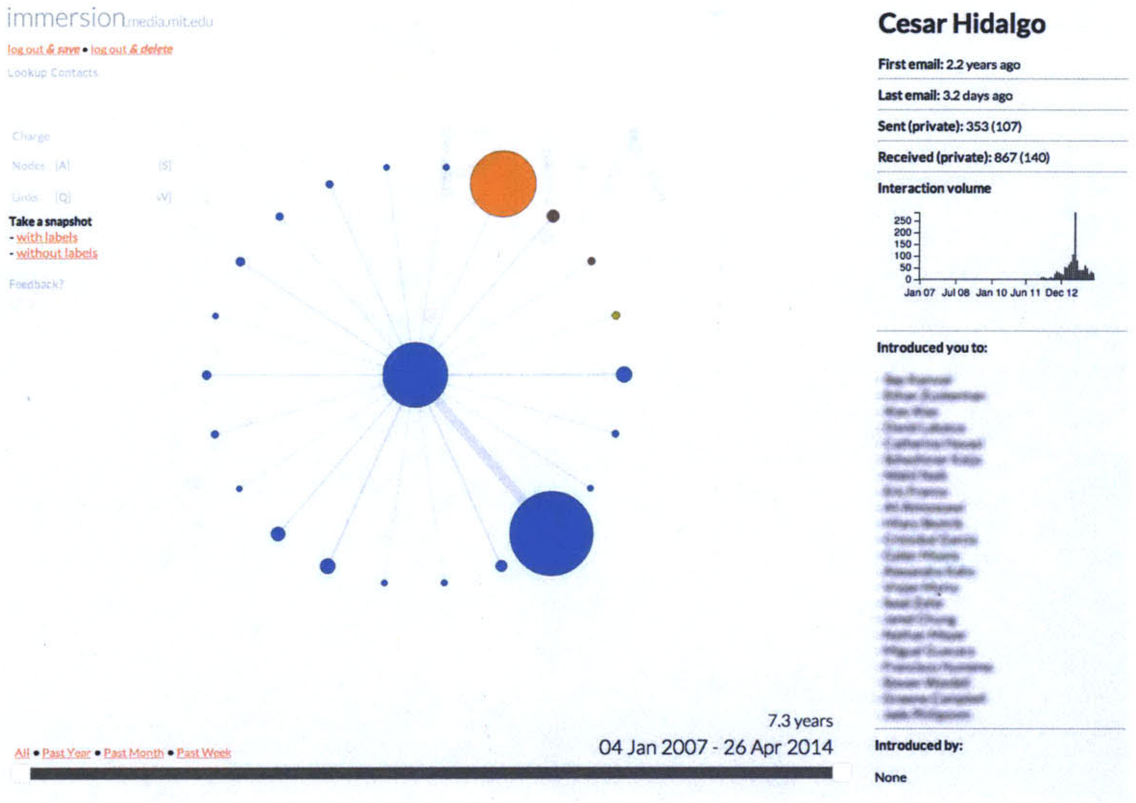


Figure 5 The network view of Immersion. The labels of the nodes are hidden to protect other's privacy, and mine.

When the user clicks on a particular contact, Immersion shows details about the relationship with the selected contact (Figure 6), such as the communication intensity over time and total number of emails exchanged. It also shows the people this particular contact has introduced the user to, as well as the person this contact was introduced by (lower right side of Figure 6). The actual names were taken out in this figure.



Cesar Hidalgo

First email: 2.2 years ago

Last email: 3.2 days ago

Sent (private): 353 (107)

Received (private): 867 (140)

Interaction volume

Introduced you to:

[redacted]

Introduced by:

None

7.3 years

04 Jan 2007 - 26 Apr 2014

Figure 6 Selecting a contact in Immersion. The list of introductions has been blurred out for privacy reasons.

Using the range slider at the bottom of the screen, the user can choose a specific time range and filter the email data. When a specific time range is chosen, the network and the list of top contacts are recomputed in real-time and shown to the user. For reference, Figure 7 shows two versions of my network, created by Immersion by using different time ranges. The two networks correspond to the time before and after joining the MIT Media Lab, shown on the left and right side of Figure 7 respectively. The orange group (in both networks) represents people in the Macedonian Academy of Sciences and Arts where I worked before coming to the Media Lab. The purple group (in both networks) shows the social network I formed during my IBM Research internship. When comparing both networks, the most apparent change is the newly formed blue group (the lower left side of the second network), which represents the people at the MIT Media Lab. The two largest blue nodes are my colleague Deepak and my advisor Cesar, respectively, and the large orange circle in both networks is my previous advisor, whose circle size decreased in the second version of the network, since my email communication focused mostly on the people at the Media Lab.

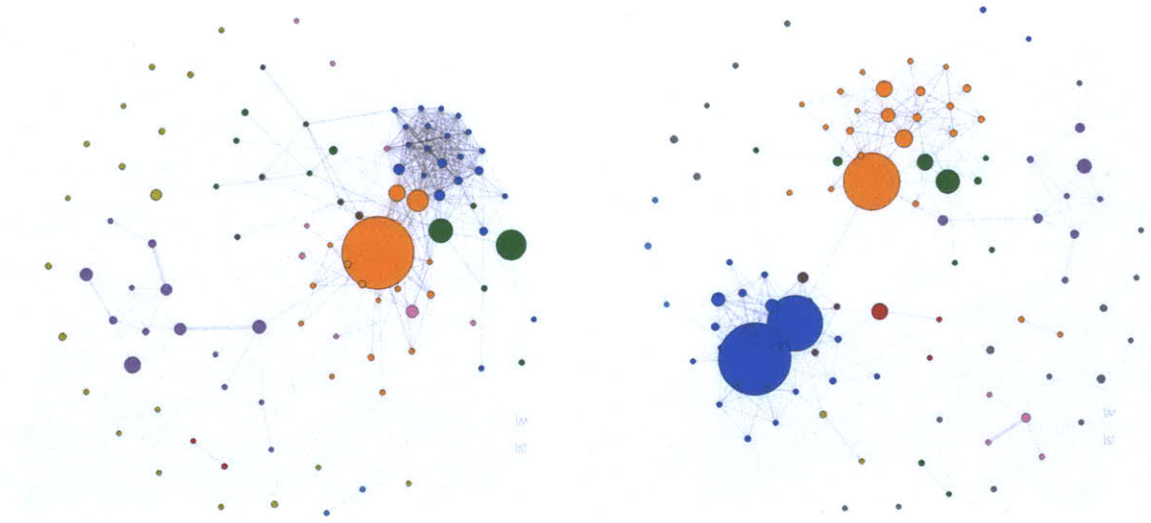


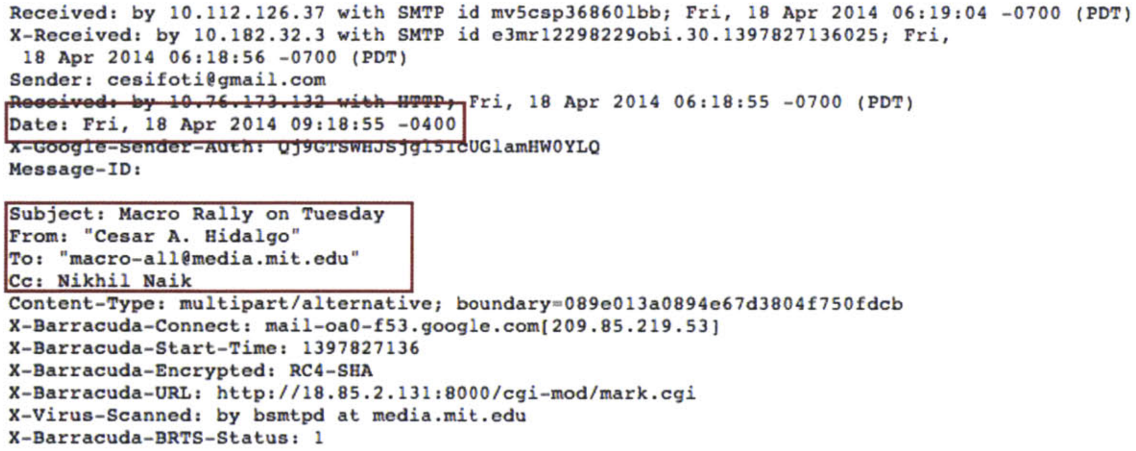
Figure 7 My email network before and after joining the MIT Media Lab.

The four aspects

Immersion is not about one thing. It is about four. It is about providing users with a number of different perspectives by leveraging on the fact that the web, and digital communication is becoming more prominent in our day-to-day communication. Immersion is also a very young project and the idea behind it, is to provide a strong framework for future interactive data analytics and visualization of not just email, but any kind of messaging data, such as Facebook, Twitter, text messages etc. To see the bigger picture, I will shortly elaborate on the four aspects of the project: privacy, self-reflection, art and strategy.

Privacy

Immersion was created with a strong emphasis on privacy. Only the `FROM`, `TO`, `CC`, `DATE`, `SUBJECT` and `THREADID` fields in the email header are accessed. Immersion transforms this raw metadata (see Figure 8) into a visual form that reveals much more than what you can see by looking at emails sequentially. One of the primary ideas behind the project was to demonstrate to the general public what others already know about them. Our support for privacy and ownership of data strongly reflects on Immersion. Users are explicitly provided with the option to delete any metadata that is collected by the tool with they logout.



```
Received: by 10.112.126.37 with SMTP id mv5csp368601bb; Fri, 18 Apr 2014 06:19:04 -0700 (PDT)
X-Received: by 10.182.32.3 with SMTP id e3mr12298229obi.30.1397827136025; Fri,
 18 Apr 2014 06:18:56 -0700 (PDT)
Sender: cesifoti@gmail.com
Received: by 10.76.173.132 with HTTP; Fri, 18 Apr 2014 06:18:55 -0700 (PDT)
Date: Fri, 18 Apr 2014 09:18:55 -0400
X-Google-Sender-Auth: Qj9GTSWHJSjgI5IcUGlamHW0YLQ
Message-ID:

Subject: Macro Rally on Tuesday
From: "Cesar A. Hidalgo"
To: "macro-all@media.mit.edu"
Cc: Nikhil Naik
Content-Type: multipart/alternative; boundary=089e013a0894e67d3804f750fdcb
X-Barracuda-Connect: mail-oa0-f53.google.com[209.85.219.53]
X-Barracuda-Start-Time: 1397827136
X-Barracuda-Encrypted: RC4-SHA
X-Barracuda-URL: http://18.85.2.131:8000/cgi-mod/mark.cgi
X-Virus-Scanned: by bsmtpd at media.mit.edu
X-Barracuda-BRTS-Status: 1
```

Figure 8 Portion of a raw email header. The highlighted parts are used by Immersion.

Self-reflection

The number of emails that we send over long period of time is orders of magnitudes larger than what we can see through our email clients. As we send and receive hundreds of emails, we leave behind unique digital traces that can reveal plenty about our personal life, even to ourselves. One of the design goals of Immersion is to allow users to reflect upon their lives, by allowing them to dive into the past, remind themselves of past interactions, and reflect on how their communication patterns changed over time.

Art

Immersion also provides an artistic representation that exists only in the presence of the visitor. A network of personal and professional contacts, free of context, has the largest value and meaning when presented to the person with the right context. In that sense, looking at Immersion sometimes feels like looking at a portrait of yourself.

Strategy

Immersion helps the users be more strategic about their professional interactions, with a map to plan more effectively whom they connect with. Moreover, Immersion can, relatively easily, be modified to look at multiple email datasets and map out the entire network of an organization or a company.

Architecture

Immersion is implemented using client-server architecture with a thin server, and a thick client that handles email parsing, data cleaning, and runs all of the algorithms for data processing and visualization. The architecture is depicted in Figure 9. All of the algorithms for data analysis, including the network extraction and clustering are written in JavaScript from scratch. The visualization code is written using the D3.js library (Bostock, Ogievetsky, & Heer, 2011). Unlike the thick client, the server is thin, acting mostly as an email downloader and a file server written in python using the Tornado web server. This architecture provided Immersion with high scalability, acting as a distributed system, since every user uses its own processing power and memory to analyze his/her own emails.

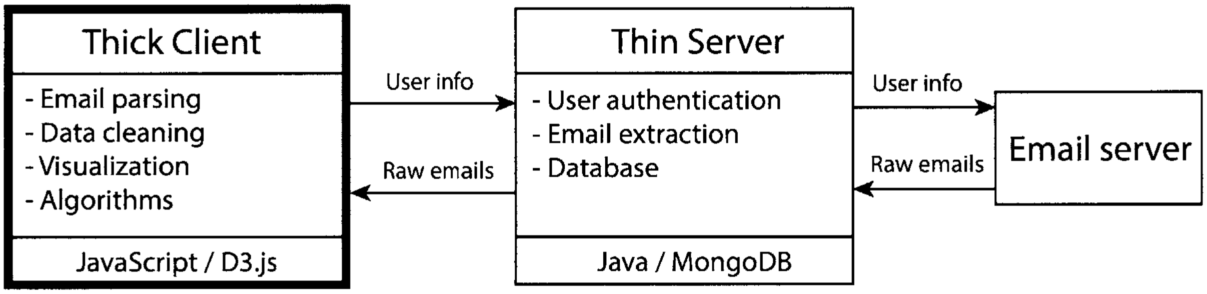


Figure 9 The architecture behind Immersion.

Upon visiting Immersion, the user provides the email login information to the server in a secure environment (the site implements an updated SSL encryption). When the server receives the login information, it creates an email extraction task, which is processed by one of the 100 threads in the thread pool. For performance reasons, we decided to implement the email extraction code in Java. Immersion supports Gmail, Yahoo and Microsoft Exchange as email providers. For Gmail and Yahoo, the email extraction is done using the standardized IMAP protocol (Crispin, 2003). For Microsoft Exchange, we use the official Java API library provided by Microsoft (Microsoft Corporation, 2011).

The extraction of emails from the email provider can take from 30 seconds to 10 minutes depending on the number of emails the user has. For security reasons, Immersion caps the number of emails to 300,000 and the total size of the emails to 50MB in compressed format. To lower the waiting time for the user and make Immersion more interactive, the extraction is implemented in batches of 10,000 emails. Whenever a batch of 10,000 emails is extracted from the email provider, the server compresses them and sends them to the client. The compression reduces the size of the email data on average by a factor of 9. This significantly lowered the time it takes for the client to download the emails from the Immersion server. When the first batch is ready, the client provides the user with the full visualization experience while letting the user know that the current visualization uses a subset of all the emails, and the email extraction process is still running.

In a data intensive application such as Immersion, caching of the data is essential. Every batch of 10,000 emails is cached both on the user's browser and in the main memory of the server, thus significantly reducing the number of hits to the hard drive of the server. The server where Immersion runs has 30GB of main memory. This allows us to store the compressed email headers in main memory for up to 3000 users at a time (assuming an average of 10MB per user).

Finally, when logging out of Immersion, the user has the option to delete or store his/her email data on our server. If the user choses to store the emails, the next time he/she comes to the site, Immersion will use incremental extraction, extracting only the emails that were created after the last visit. This incremental extraction usually takes less than 10 seconds providing seamless experience for the user.

Data cleaning

Before doing any kind of computation, Immersion preprocesses the raw email data received from the email server in order to provide consistency across all records. Data cleaning (Rahm & Do, 2000) is essential since a marginal increase in the quality of the data can provide significantly better analytics and visualization results later in the pipeline.

The first step is parsing the email. I will skip explaining the parsing process, however, because the only non-trivial step is the parsing of dates (different email clients use different formats to describe dates). After parsing, Immersion eliminates emails with invalid dates, such as dates in the distant past or the future, as well as an empty `FROM` header. An empty `TO+CC` header is not a problem, since every received email has at least one recipient, the actual user. An empty list of recipients could mean the user was placed in the `BCC` field. To avoid complications, Immersion always removes the user from recipients list and assumes he/she is always in the list.

Another complication involves matching people with multiple email addresses (or aliases). Initially, Immersion detects all of the email addresses used by the user of the tool by going through the "sent" email set (emails in the `SENT` folder) and collecting the unique email addresses in the `FROM` header. Then, Immersion looks at the "received" email set, in order to detect other "sent" emails by matching the `FROM` header with the user's aliases. This is important since when people migrate emails from one to another email account, they don't necessarily put the "sent" emails in the `SENT` folder. For the rest of the contacts in the user's ego-network, Immersion assumes that there are no two people that share the same name and last name. With this assumption, a unique contact is created for each unique name normalized to the canonical format "FirstName LastName". This normalization is important since the original name can be in the format "LastName, FirstName". Moreover, names can contain titles (e.g. M.D., Ph.D.) or alternative names like middle and maiden names, which are sometimes enclosed in parenthesis.

Not every contact is represented in the network. Some email addresses belong to a mailing list or a promotion list (e.g. Facebook and LinkedIn email updates) and Immersion ignores these contacts. A simple heuristic that works well is to assume that contact i is a real person if we have both sent and received at least k emails with i. The reason this works well is that in the case of a mailing list, we always send to a mailing list, but the mailing list is never the sender of the email. On the other hand, we receive emails from promotion lists, but we almost never reply to those emails. Sometimes there are exceptions, and to handle these exceptions, we use $k = 2$ which was found to work well empirically. The contacts that pass the real person test are referred in Immersion as *collaborators*.

After the mapping of email addresses to collaborators, each email record is normalized by combining the `TO` and `CC` headers into a `recipients` field (treating the `TO` and `CC` headers equivalently), and replacing the email addresses with the id of the corresponding collaborator. Finally, automatically created emails, such as auto replies, are detected and flagged by looking at the `Auto-Submitted` header.

Network extraction

Our social interactions, both on a professional and personal level, display patterns of connections that are neither regular nor random. A whole field called network science is dedicated to study these patterns (Albert & Barabási, 2002; Newman, 2003). Network science gave birth to many methods to help us understand the complexity of these networks and these methods play a large part of the computational machinery behind Immersion.

The main visualization of Immersion is the network view. Each contact is represented as a circle and the radius of the circle is proportional to the communication strength between the user and that contact. The communication strength of contact i, α_i, is calculated as the generalized mean of the number of emails that the user has sent to i, s_i, and the number of received emails from i, r_i:

$$\alpha_i = \left[\frac{s_i^p + r_i^p}{2}\right]^{\frac{1}{p}}.$$

Initially, we used $\alpha_i = \min(s_i, r_i)$, which corresponds to $p = -\infty$, however we found empirically $p = -5$ to work well for most cases. This gives more strength to a two-way (symmetric) communication than a one-way (asymmetric) communication for the same total number of exchanged emails, while still considering the total number of exchanged emails.

Interaction ties between two contacts are represented as lines between circles and the thickness of the line corresponds to the communication strength between the two contacts. These ties are inferred by looking at emails with 2 or more participants, not counting the user itself. For each pair of users i and j, we will denote the number of emails sent from i to j as e_{ij}. We then define the strength of the interaction between users i and j, β_{ij}, in similar fashion as before, as the generalized mean between e_{ij} and e_{ji} with $p = -5$. Note that since we are only looking at one user's mailbox, the communication strength between the users i and j is according to that user's perspective and doesn't necessarily correspond to the general communication strength between i and j.

Network visualization

Immersion uses d3.js (Bostock, Ogievetsky, & Heer, 2011), a JavaScript library for interactive data visualization. The network layout is calculated using a modification of the d3's build-in force-based layout algorithm. The basic idea behind force-based layouts is that every node has a repulsive charge force, which keeps nodes away from each other, while every link between nodes acts as a spring force, keeping linked nodes together. Additionally, there is a pseudo-gravity force originating from the center of the layout. This force avoids expulsion of disconnected subgraphs by keeping nodes close to the center of the layout. For a more realistic layout, each node i has a charge force proportional to α_i, and each link between contacts i and j has an associated spring force proportional to the interaction strength β_{ij}. Finally, there are bounding box constraints to keep the nodes confined within the boundaries of the canvas.

Community detection

A well-known phenomenon in social networks, and networks in general, is the division of the network into communities (also called groups, clusters or modules). The problem of automatically detecting these communities has been well studied and is known as the community detection problem. For a good overview of the community detection problem see (Fortunato, 2010). The basic idea behind a community is that there is a higher density of links within communities than between them. The most popular methods for community detection are based on the modularity quality function (Newman & Girvan, Finding and evaluating community structure in networks, 2004). Given a partition of nodes in the graph, the modularity function tells you the quality of that partition. Intuitively, the modularity function gives high value to partitions that have many links within communities and only a few between them. Thus, the main goal of all the modularity-based algorithms is to maximize the modularity function by constantly moving nodes from one group to another in order to achieve a "cleaner" community split (Clauset, Newman, & Moore, 2004; Duch & Arenas, 2005; Newman, Modularity and community structure in networks, 2006).

The initial community detection algorithm in Immersion was Infomap (Rosvall & Bergstrom, 2008), which is not based on the modularity quality function, but on the idea of compressed random walks on networks. However, later we decided to switch the implementation to a modularity-based algorithm (Clauset, Newman, & Moore, 2004) due to its simplicity and comparable performance given the relatively small size of the networks.

Let us describe the basic idea behind the modularity function. Let A be the adjacency matrix with $A_{ij} = 1$ indicating that nodes i and j are connected ($A_{ij} = 0$ indicating an absent connection). Also, let the community of node i be denoted as c_i. Then the fraction of links that connect nodes in the same community is given by community density function

$$C = \frac{\sum_{ij} A_{ij}\delta(c_i, c_j)}{\sum_{ij} A_{ij}} = \frac{1}{2m}\sum_{ij} A_{ij}\delta(c_i, c_j),$$

where $\delta(i, j) = 1$ if $i = j$ and 0 otherwise and m is the total number of links in the graph. C is large when links connect primarily nodes within the same community. Yet, the community density it is not ideal on its own, since it takes its largest value of 1 in the trivial case when all nodes belong to a single community. The idea behind the modularity function is to take the community density C and subtract the expected value of the same quantity when the network is randomized. In a random network where everything is randomized except the degree of the nodes (the number of links adjacent to a node), the probability of a link between nodes i and j is given by $\frac{k_i k_j}{2m}$ where k_i is the degree of node i. Then the modularity function is defined as

$$M = \frac{1}{2m}\sum_{ij}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta(c_i, c_j).$$

Modularity M is zero when the fraction of links within communities is no different than what we would expect for a randomized network. A significantly positive value indicates that the network has a non-trivial modular organization.

Finding the partition with the highest modularity value is hard, since the number of possible partitions grows exponentially with the size of the network. Most algorithms use greedy techniques to optimize the modularity M, starting with each node being in its own community and then repeatedly joining two communities whose composition increases the value of M.

Following the instructions of (Clauset, Newman, & Moore, 2004), we implemented from scratch in JavaScript an efficient algorithm of the previously described technique, which I will briefly elaborate here. Let ΔM_{ij} be the change in M that would result from joining the two communities i and j into a single community. Then, the basic idea across most of the greedy algorithms is to find the pair i, j with the largest ΔM_{ij} efficiently without computing ΔM_{ij} for all possible pairs i, j. The main insight is that joining two communities with no link between them will never produce an increase in M, thus, we only need to compute ΔM_{ij} for pairs i, j that are joined by one or more links. Additionally, an efficient data structure is used to keep track of the largest ΔM_{ij} as the algorithm progresses. The data structure is a combination of a sparse matrix that holds ΔM_{ij} for each pair i, j with at least one link between them and a max-heap, which contains the largest element of each row of the sparse matrix. Moreover, each row of the space matrix is stored as a balanced binary tree allowing retrieval and insertion of elements in $O(\log n)$ time. This custom data structure allows performing updates quickly when joining two communities. The running time of this algorithm for a graph of n nodes and m links is $O(md \log n)$ where d is the depth of the dendrogram describing the community structure. Since many real-world networks are sparse and modular, i.e. $m \sim n$ and $d \sim \log n$, in practice, the algorithm runs close to linear time, $O(n \log^2 n)$. For more details about the algorithm, I refer the reader to (Clauset, Newman, & Moore, 2004).

Basic statistics

One of the statistics Immersion computes is the number of emails sent, emails received and new collaborators acquired over time. These statistics are presented using a histogram bar chart with one bar per year (see Figure 10). The last bar of the histogram is a stacked bar, containing a black and a grey rectangle. The grey rectangle is the prediction Immersion makes for the rest of the current year, obtained by taking an average between the last year's and the current year's trend.

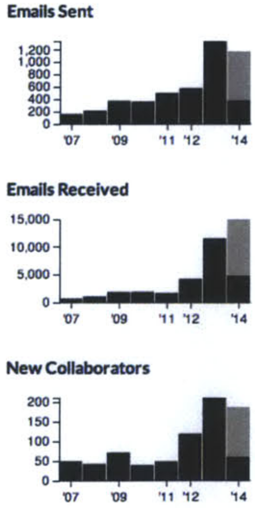


Figure 10 Histogram showing the number of sent and received emails, as well as the number of newly acquired collaborators for each year.

Immersion also gives you a list of the top collaborators sorted by the communication strength α_i. When the user hovers over a collaborator, the node and its links are immediately highlighted in the network as shown in Figure 11. The user can also search for a contact by typing the name of the contact in a search box.

When selecting a collaborator A, the user is shown a list of the collaborators that were introduced by A, as well as the collaborator that introduced A, if there is one. We say that collaborator A introduced collaborator B to the user if the first time the user ever saw *B*'s email address was when *B* was in the list of recipients in an email sent by *A*.

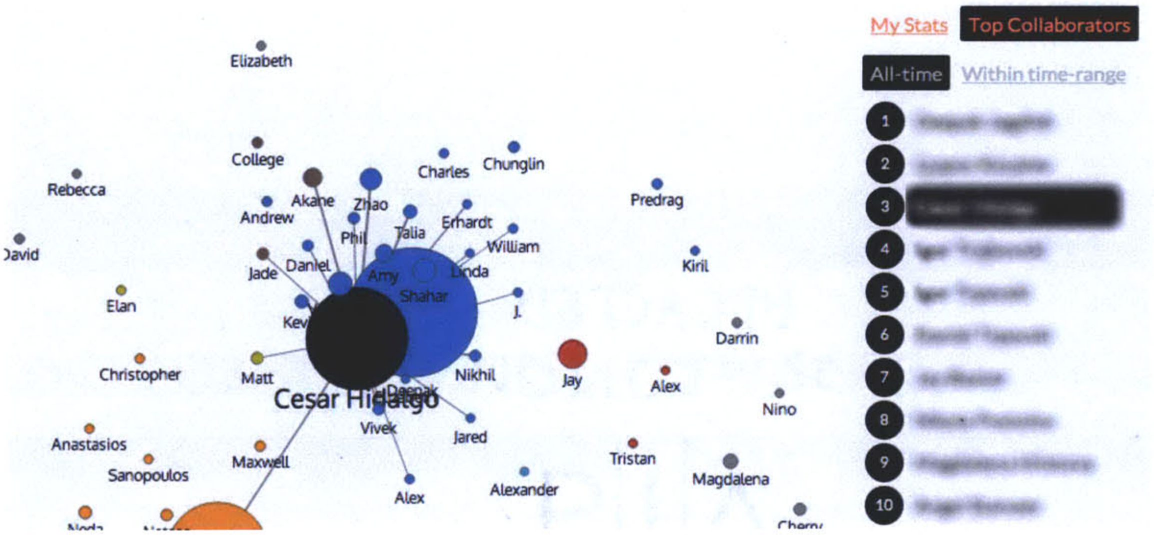


Figure 11 When hovering over a collaborator, the corresponding node in the network and its associated links are highlighted.

Impact

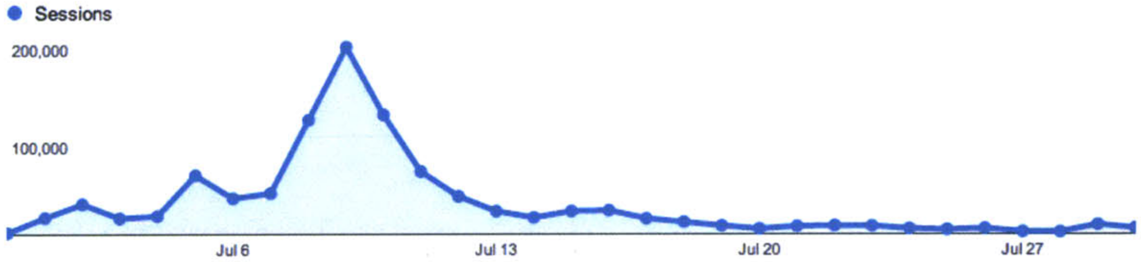
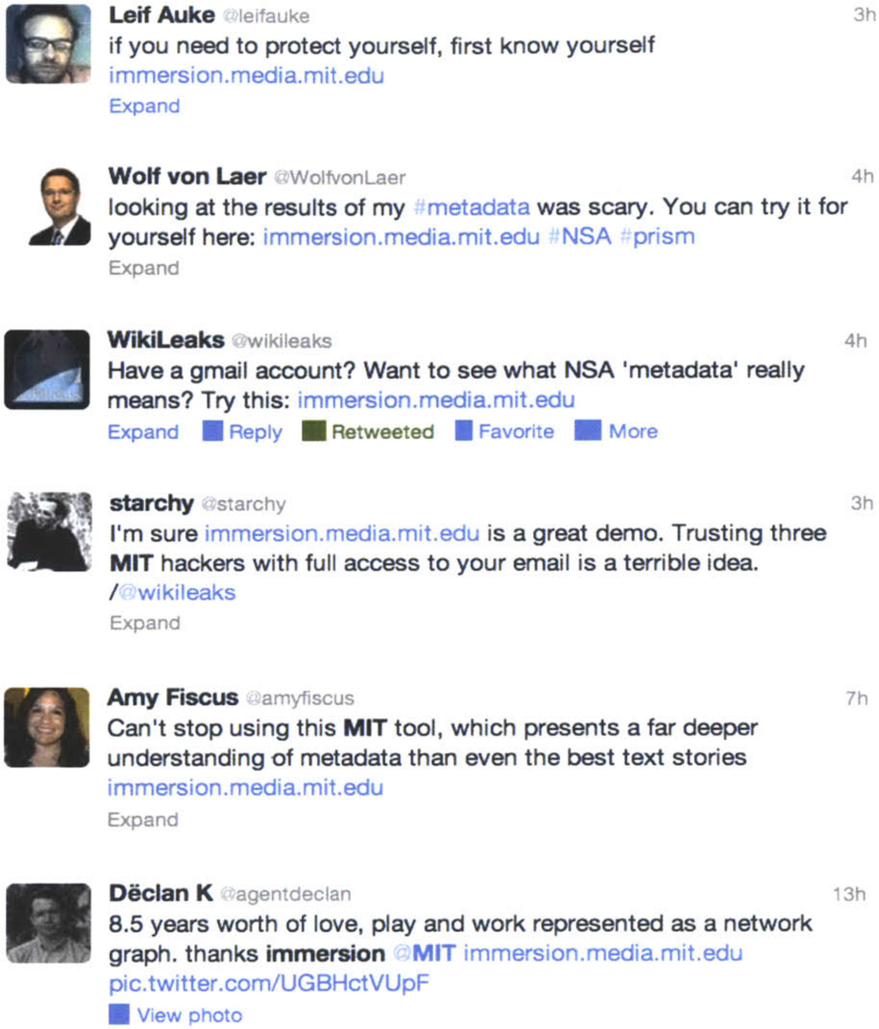


Figure 12 Number of visits per day for the first month after the launch of Immersion.

Immersion was officially launched on June 30, 2013. The launch was covered by a story in the Boston Globe titled "What your metadata says about you" (Riesman, 2013). The day after the launch, NPR also wrote a story about Immersion, again taking the privacy perspective, describing an MIT tool that, in the author's words, let's you spy on yourself (Goldstein, 2013). At this point we already knew that the privacy and metadata aspect of Immersion would make the strongest impression on the general public, especially since the launch coincided with the peak of the NSA debate and the Edward Snowden story.

The NPR story quickly became the most popular story on the site, generating around 30,000 visits in a single day. We were not ready for such traffic and our server crashed due to scalability issues. By July 2nd, the service provided by Immersion was shut down, and we left a message on the Immersion website that we are coming back soon. The public's attention didn't go down however. Instead, the number of news articles, blogs and tweets written about Immersion was increasing every minute, resulting in about 5,000 people signing up in the mailing list to be notified when the service is going back online. After a 3-day coding marathon, the non-scalable parts of Immersion were re-implemented resulting in a re-launch on July 4th. By July 7th we had approximately 205,000 unique visitors with 320,000 visits and over a million page views. The peak day however, was July 9th, with almost 200,000 visits (see Figure 12) in that day. At that point major US news sites such as New York Times, Time Magazine, Wired, Forbes etc. and international news sites, such as The Guardian, Le Monde, La Repubblica, Deutsche Welle etc. wrote an article about Immersion. Sharing on social media also generated a lot of visits. Figure 13 shows some of our user's reactions. Since its launch, until May 1, 2014, Immersion has accumulated over 750,000 unique visitors with over 1.3 million visits, with 43% returning visitors. In the last half a year, Immersion's traffic stabilized with approximately 9000 unique visitors per month.



Leif Auke @leifauke 3h
if you need to protect yourself, first know yourself
immersion.media.mit.edu
Expand

Wolf von Laer @WolfvonLaer 4h
looking at the results of my #metadata was scary. You can try it for yourself here: immersion.media.mit.edu #NSA #prism
Expand

WikiLeaks @wikileaks 4h
Have a gmail account? Want to see what NSA 'metadata' really means? Try this: immersion.media.mit.edu
Expand Reply Retweeted Favorite More

starchy @starchy 3h
I'm sure immersion.media.mit.edu is a great demo. Trusting three **MIT** hackers with full access to your email is a terrible idea. /@wikileaks
Expand

Amy Fiscus @amyfiscus 7h
Can't stop using this **MIT** tool, which presents a far deeper understanding of metadata than even the best text stories
immersion.media.mit.edu
Expand

Dĕclan K @agentdeclan 13h
8.5 years worth of love, play and work represented as a network graph. thanks **immersion** @**MIT** immersion.media.mit.edu
pic.twitter.com/UGBHctVUpF
View photo

Figure 13 Some of the Immersion users reactions on Twitter

3. IMMERSION V2

In the next iteration of Immersion, we introduced several new features. One of them is the query engine that provides extensive filtering capabilities by allowing the user to quickly filter and slice email data by different dimensions. Additionally, we added language analysis by including the subject of the email, which is also part of the email header. Consequently, we added a custom word cloud visualization of topics related to the results of the query engine to quickly give the user an idea behind the content of the related emails. Moreover, we introduced several improvements of the network layout such as non-overlapping node labels and zooming capabilities. Finally, we worked hard to achieve near real-time response, obtaining results from hundreds of thousands of emails in the order of tens of milliseconds, giving the user an experience of a highly reactive and interactive data exploration engine.

Data cleaning and indexing

With the addition of email subjects, Immersion needs to perform some basic language preprocessing. The initial step in the language analysis is to avoid repetition of subjects within the same email thread. In Immersion, we refer to an email thread as *conversation*. To do this, we only look at the level of conversations and associate each conversation with the subject of the first email within that conversation. Then we filter out action words like "Fwd:" and "Re:" from the subject. After that, we tokenize the subject into set of words, and apply the Porter Stemming algorithm (Porter, 1980), which converts the word into a normalized form by stripping the suffixes. For example, the words "algorithm", "algorithms" and "algorithmic" are going to be converted to the word "algorithm". This step is important because the importance of a word, which we refer to as *topic* in Immersion, is determined by counting the number of occurrences of a given word, and we want to keep a single counting bucket per concept. The importance of a topic w, β_w, is calculated in similar fashion as the importance of a collaborator:

$$\beta_w = \left[\frac{s_w^p + r_w^p}{2}\right]^{\frac{1}{p}},$$

where s_w is the number of conversations initiated by the user (the user writes the subject) that include word w. Similarly, r_w is the number of conversations initiated by someone else that include the word w. Empirically, we found $p = -5$ to work well in most cases.

To achieve real-time interactive data exploration, we need to be able to search and filter through hundreds of thousands of emails within a few milliseconds. To achieve this design goal, we built a custom in-memory index, with four basic types of objects: email, collaborator, topic, and conversation, with pointers between them for fast querying. More concretely, each email, besides the basic email information, has a pointer to the conversation object. Each conversation object contains the email subject, pointers to the other emails in the same conversation, and pointers to the topic objects that are related to the conversation. The collaborator object on the other hand, has pointers to all the emails that include him/her. This enables quick finding of the emails that include a specific set of contacts. Analogously, the topic object contains pointers to the conversations that include that topic, allowing quick retrieval of the set of conversations that include a specified list of topics.

Finally, every part of the interface in Immersion is updated by providing a filtered set of emails, which are obtained using the in-memory index and a custom filtering procedure. The time it takes to apply the filtering procedure is in the order of tens of milliseconds. Figure 14 shows an example of the filtering process. The basic idea is to iteratively compute a new filtered set of emails by taking an intersection between the current email set and the email set obtained using the in-memory index by taking one of the constraints in the filter. The intersection generally returns a much smaller set of emails. The time range filtering is applied at the end of the filtering process. Since all intermediate sets of emails are sorted by time, the time range cutoff is determined by using a binary search algorithm.

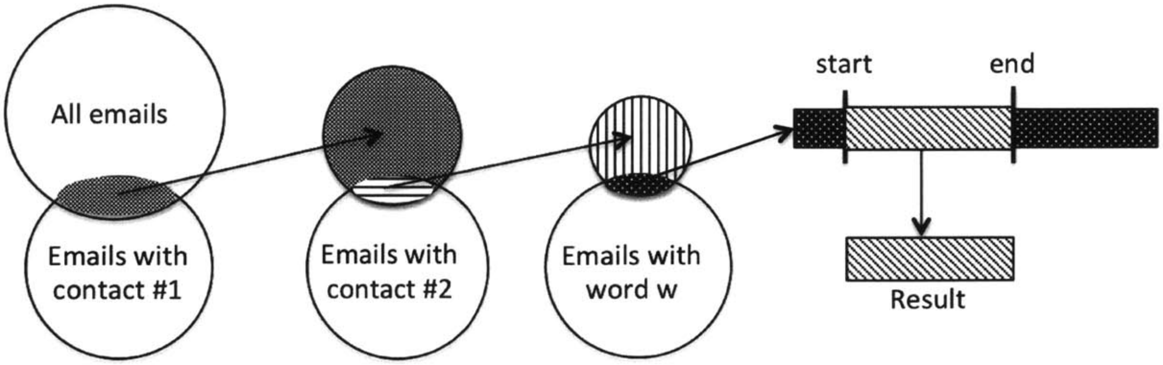


Figure 14 The filtering process in Immersion. The result contains the emails whose dates are between *start* and *end* date and include contact *#1*, contact *#2* and the word *w* in the subject.

Design

The design of the second version, shown in Figure 15, was similar to that of version one, keeping the network layout central to the interface. While the first version of Immersion focused on the individual, the new version focuses on querying and filtering emails in order to obtain a smaller and more meaningful subset of emails, which drives all of the other parts of the interface. In this sense, Immersion acts as a database query engine, providing extensive filtering capabilities. The user can filter emails by three different types of items: time, people and topics. The filter can also contain multiple items of the same type (e.g. multiple people), as well as a combination of the different types. This allows diving deep into the email dataset and looking at different slices of the data.

Let's start by explaining the interface in more detail. The timeline on the bottom of the screen was improved to provide a histogram showing the total number of emails exchanged over time. Above the timeline is the filter description, which shows the current items in the filter and the number of conversations in the current filtered set of emails. On the left side, Immersion provides a list of topics sorted by a topic importance factor. On the right side is the list of collaborators, sorted by the communication strength α_i. The central part of the interface depends on the selected view. All parts of the interfaces are reactive, i.e. their content changes dynamically based on the filtered set of emails. There are 4 different types of views: the original network view, the stats view, the topics view and the subjects view. The stats view provides basic statistics that are derived from the filtered set of emails. The topics view provides a coordinate sensitive word cloud constructed from the topics of the emails. The subjects view gives the user the full subject of the emails, and upon clicking on a subject, the user is taken to the original Gmail thread associated with that subject. For now, this new capability of linking to the actual email threads works only for Gmail, but we plan to add support for all of the email providers that were supported in the first version.

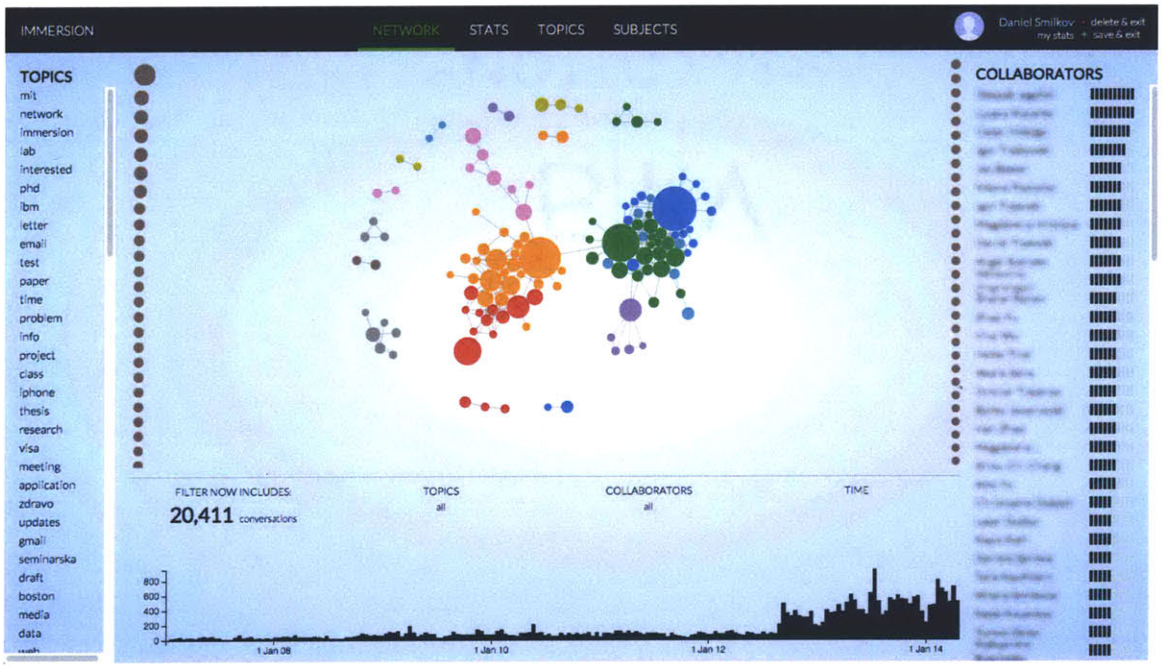


Figure 15 The new interface of Immersion powered by my email metadata. The names of the collaborators have been blurred out and the node labels in the network are hidden for privacy reasons.

The network layout was improved by adding collision detection in the force-based layout simulation so that nodes don't overlap with each other. Additionally, nodes with no links are removed from the actual simulation and shown on the side. This makes the network visualization cleaner and avoids misinterpretation of the x and y coordinates of those nodes within the context of the larger network. The user can zoom in and out of the network, and the labels of the nodes are dynamically hidden to avoid label overlap, with the labels of the larger nodes having more priority in being shown. Another difference is that the network layout is fixed and only the collaborators related to the current filtered set of emails are highlighted.

Reactive interface

When the user hovers over a topic, a collaborator, a bar on the histogram, over a node in the network or a topic in the word cloud, several things happen:

- The corresponding filter item gets added to the temporary filter.
- The new filtered set of emails is computed.
- All necessary computations that fuel the rest of the interface are performed again.
- The interface gets updated. The list of topics and collaborators changes. Part of the email histogram is highlighted according to the new filtered set of emails (Figure 16).

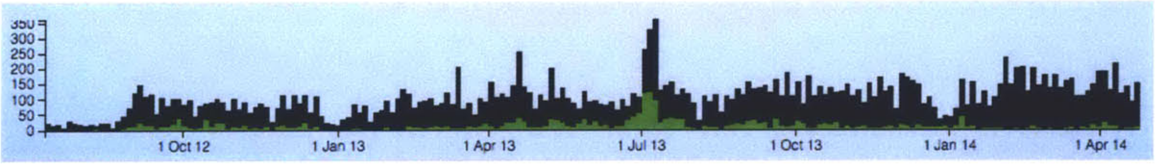


Figure 16 When a temporary filter is applied, part of the email histogram corresponding to the new filtered set of emails is highlighted.

As mentioned previously, all of these steps are performed in the order of tens of milliseconds. This fast computation time was a major design goal in the new version in order to provide real-time interactive data exploration.

When the user hovers out of the item, the item is removed from the temporary filter. This allows the user to quickly explore the dataset and compare different items. An example includes hovering over the list of collaborators, and quickly seeing the related topics for each user, as well as the co-collaborators shared with that user by looking at the highlighted nodes in the network view (see Figure 17).

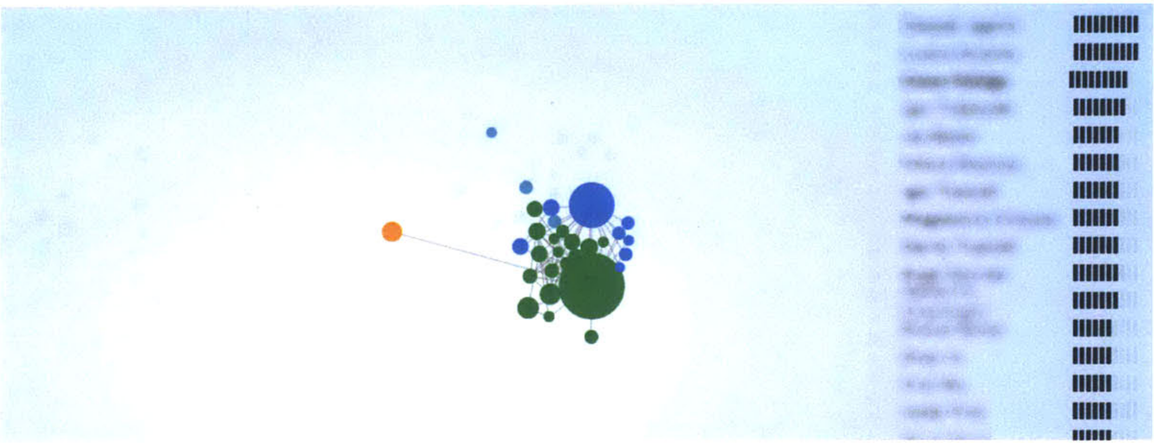
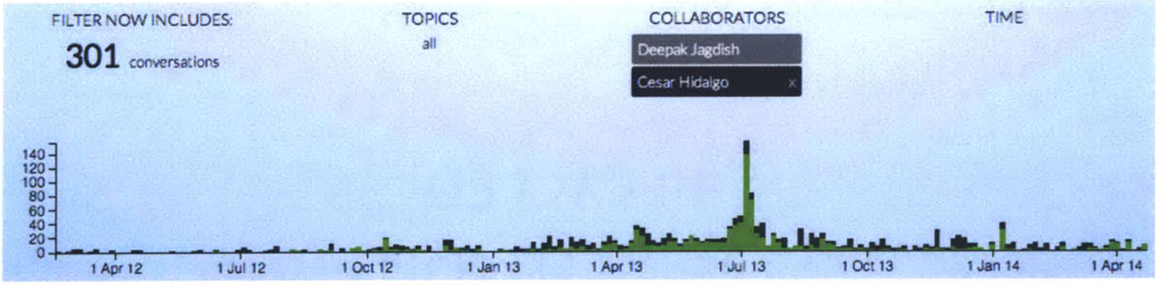


Figure 17 Hovering over a collaborator. The network layout stays fixed, while the shared collaborators are highlighted in the network.

When the user clicks on an item, the item gets added to the permanent filter. This means that when the user hovers out of the item, the item will stay in the filter. Then the user can explore the new slice of the data, add new filters in the same way as before, and go deeper into the data.



FILTER NOW INCLUDES:
301 conversations

TOPICS
all

COLLABORATORS
Deepak Jagdish
Cesar Hidalgo x

TIME

Figure 18 The filter description along with the timeline. My advisor Cesar has been permanently added to the filter, while I'm hovering over my colleague Deepak, which adds him to the temporary filter. The filtered set of emails contains 301 conversations that have both Deepak and Cesar in the conversation. One can quickly see that my communication with Deepak is highly correlated with that of Cesar, pointing to the fact that when Cesar and I email each other, we almost always add Deepak to the list of recipients. The peak in the beginning of July of 2013 corresponds to the launch of Immersion.

Topics view

The idea behind the topics view is to allow the user to quickly grasp the content behind the email conversations. It is based on a compact visual form of words, commonly known as the word cloud. Word clouds are a visualization method where the image is composed of words used in a particular text, in which the size of each word indicates its importance. In recent years, word clouds have been widely used to provide content overview of a website or a set of documents. The problem with regular word clouds, however, is that the position of the words is randomly chosen, and words that frequently co-occur are not necessarily close to each other. There has been some recent work that overcomes this problem by allowing the user to provide a similarity matrix indicating the similarity between every pair of words (Cui, 2010; Hassan-Montero & Herrero-Solana, 2006). I refer the reader to (Viegas, Wattenberg, & Feinberg, 2009) for details concerning the usual layout algorithms used to determine the position of the words in the word cloud. In Immersion, however, we take a different approach. To determine the coordinates of each word, we use a force-based layout algorithm with each node corresponding to a word, and the strength of the spring force acting on each link being proportional to the similarity between the associated pair of words. To determine the similarity of the pair of words A and B, we use the relative co-occurrence, as measured by the Jaccard index (Han, Kamber, & Pei, 2006):

$$J(A,B) = \frac{|S_A \cap S_B|}{|S_A \cup S_B|},$$

where S_A and S_B is the set of emails whose subjects contain the word A and word B respectively. To avoid overlapping of words, the algorithm iterates over each word by decreasing importance and shows the word only if it doesn't overlap with any of the previously shown words. A word cloud constructed from my emails exchanged with my advisor Cesar (he is added to the filter) is shown in Figure 19.

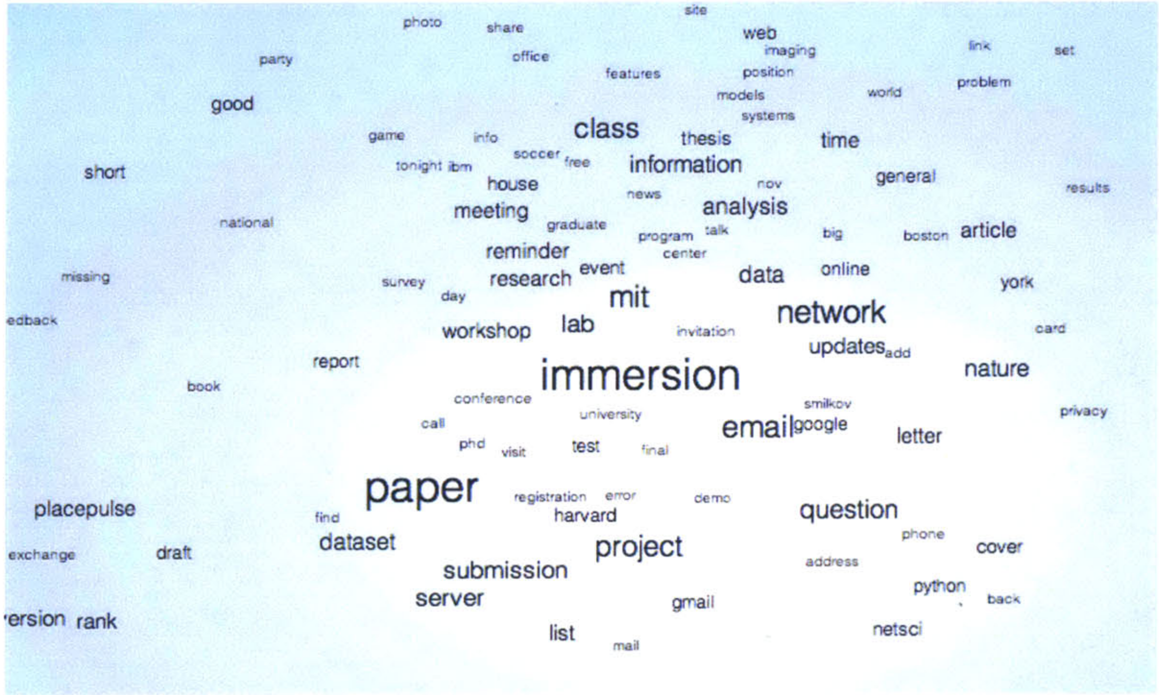


Figure 19 Word cloud based on subjects of emails that I exchanged with my advisor Cesar.

Moreover, the user can zoom in and zoom out of the word cloud, adding depth as an additional dimension. When zooming in, words with lesser importance that were not shown before are starting to appear (see Figure 20).

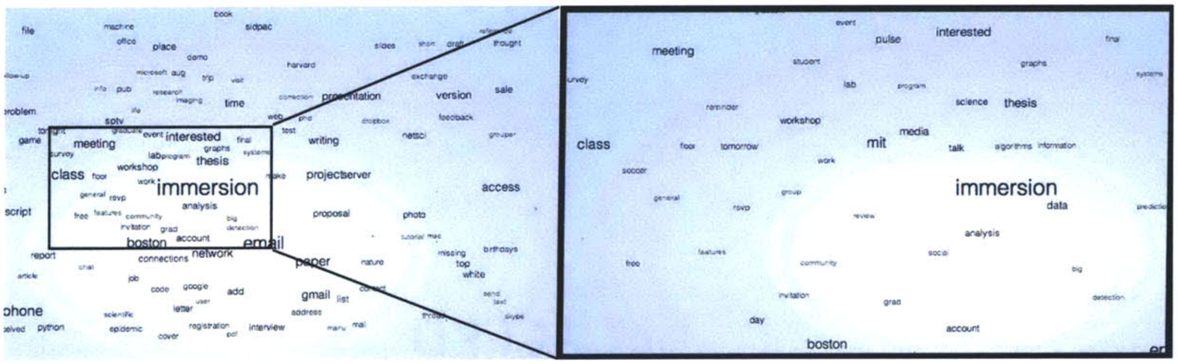


Figure 20 Before (left) and after (right) the user zooms in a particular region of the word cloud. Some words that are shown on the right side were hidden before zooming in to prevent overlap.

4. EMAIL TRENDS

As previously mentioned in the Impact section in Chapter 2, since Immersion's launch in June 2013 to this day, we have received around 750,000 unique visitors and more than 1.3 million visits. Out of the 750,000 visitors, over 180,000 kept their email headers on our servers, providing us with a longitudinal dataset of approximately 4.7 billion email headers spanning 8 years. This chapter presents some initial exploratory data analysis on the longitudinal dataset. The analysis is still at an early stage, but the idea is to look at the data and try to learn something about our email patterns, and hopefully, about human online communication in general.

In particular, we will look into global email trends such as the email overload problem (Whittaker & Sidner, 1996; Fisher, Brush, Gleave, & Smith, 2006) by measuring the change in the email volume per person, as well as the effect it has on our performance to reply to emails. We will also look at work-life balance trends by measuring the fraction of emails that were sent on a weekend.

Biases and limitations

Before we delve into the results of the email analysis and draw conclusions, it is necessary to discuss some of the biases and limitations of the data that was collected.

One of the biases is the audience behind Immersion. The average user of Immersion is probably more familiar and confortable with using technology and email than the average person. While this bias might affect absolute numbers for general email usage, looking at changes through time instead of absolute numbers can alleviate the effect. Also, I would like to note that more than 95% of the dataset comes from Gmail users. Another possible bias can come from the fact that users who shared their data with us, behave somewhat differently than users that deleted their data after the Immersion experience. Nevertheless, it is hard to measure the effects of this type of bias.

One significant limitation of the collected dataset is the lack of email subjects. This limitation is by design, since the data comes from users using the first version of Immersion. The launch of the new version of Immersion will collect email subjects from our users, which would open new avenues for analysis using techniques in natural language processing. Another limitation is the noise and inconsistency of the data. It is well know that the majority of email users use more than 1 email account. More accurately, according to a recent study, the number of email accounts per user is 2.9 (The Radicati Group, 2013). At the time of writing, Immersion doesn't provide the user with an interface to add multiple email accounts to his/her profile. Note however, that providing the service of multiple email accounts doesn't guarantee that the user will explicitly login and choose to store all of its email accounts on our servers. Nonetheless, before doing the data analysis, we try to automatically discover all the email aliases of a given user and combine the emails into a single account.

System for interactive data analysis

When doing exploratory data analysis, it is essential to get from analysis to results quickly. With this in mind, we developed an interactive program that keeps a sample of the database in main memory and dynamically loads new analysis code and executes it across multiple processing cores, obtaining results in the order of minutes. To reduce computation time, we limited our analysis to a random sample of 15,000 people containing around 415 million email headers (~9% of the dataset). The size was determined so that the sample can fit in the main memory of the machine (64GB) where the analysis was performed. To test for sufficiency of the sample size and confirm robustness of the results, the same analysis was performed on two smaller samples, containing the emails headers of 5,000 and 10,000 people respectively.

For the purposes of doing exploratory data analysis, we avoided joining data of multiple users, treating each user's emails as a separate dataset. This allowed us to assign a parallelizable map task for each user. Upon initialization, the python program reads the entire sample in main memory and has the capability to dynamically load new code for the mapping and the aggregation and visualization task. When the new code is loaded, it executes the map task for each user in parallel utilizing all of the cores of the machine and aggregates and visualizes the results. Ignoring the long initialization time of 15 minutes, which happens only once when the sample is loaded in the main memory, the time for executing new analysis code averaged around 2 minutes.

Email trends

General email volume

To see the overall email usage trend in the past several years, we compute the number of received and sent emails for every month (and year) between January 2006 and May 2013 for the median and the top 1% email users respectively. In computing the median and the 99% percentile for a particular month, (year) we only include users that have used Gmail at least a year before that particular month (year). This condition alleviates the bias from having lower usage rate in the past because of non-active users. The results, aggregated by month and year, are shown in Figure 21 and Figure 22, respectively.

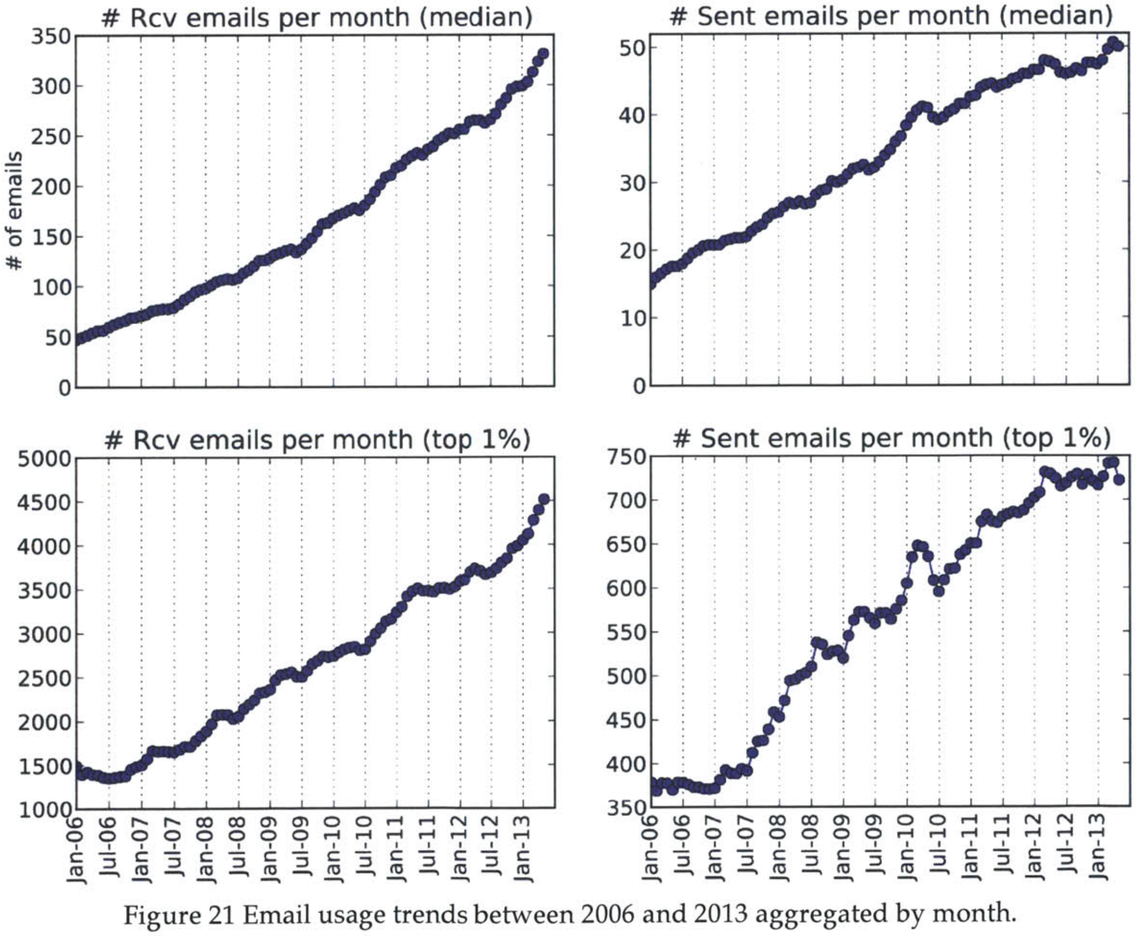


Figure 21 Email usage trends between 2006 and 2013 aggregated by month.

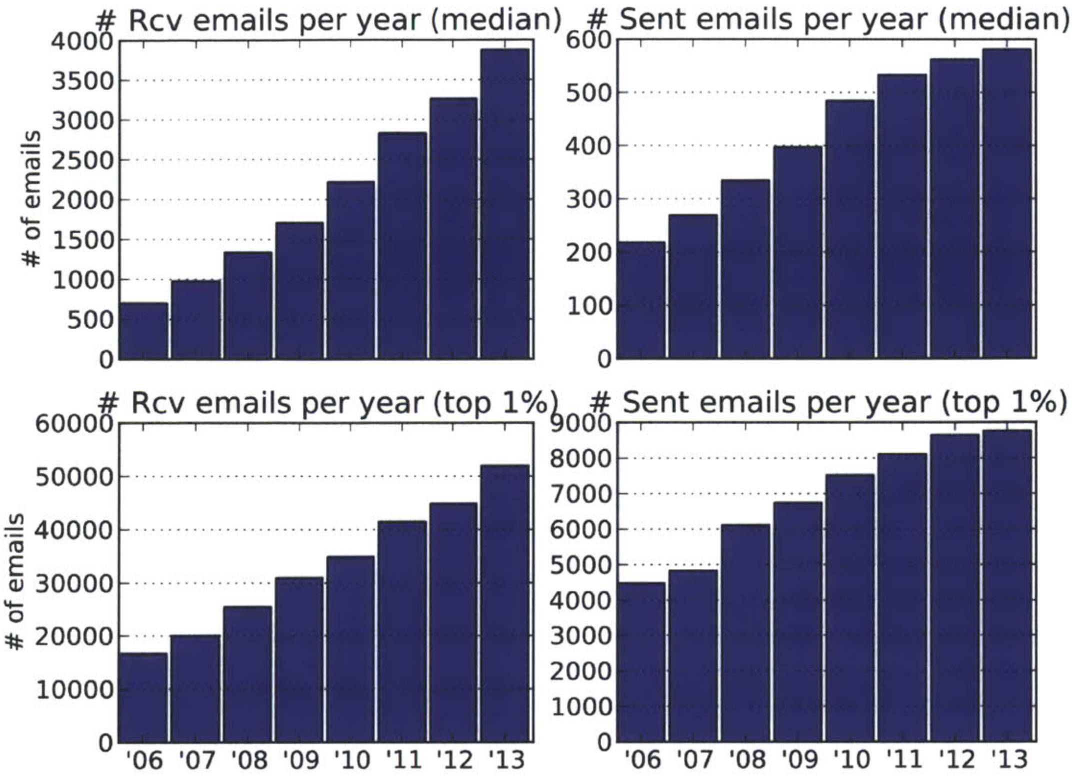


Figure 22 Email usage trends between 2006 and 2013 aggregated by year.

The increase in email usage in the past 7 years is surprisingly large, pointing to the email overload problem. The median user in the beginning of 2006 received 50 emails per month as opposed to 330 emails per month in May 2006 (6.5 times more). The number of sent emails has a slower growth, starting with 16 emails per month in 2006 and increasing to 50 emails in 2013, which is 3.1 times more. The different increase rates between sent and received is expected since received emails include machine-composed emails whereas the sent emails are emails sent by real people, obtained by looking in the SENT folder of real Gmail users. It is interesting that the number of sent emails per month for the top 1% senders has plateaued in the past 1.5 years at around 730 emails per month, most likely because of time and cognitive limits of humans. The number of received emails for the top 1% receivers, however, which includes broadcast and machine-composed emails, is nowhere near convergence; we are subscribing to more and more information.

Reply time of emails

While the significant increase in the number of emails we sent in the past several years is a strong indicator for the increasing time we spent on processing email, it is interesting to see if this has impacted our ability to reply to emails quickly. To see this, we measure the reply times of emails. This was possible to compute because we had the THREADID of every email. Emails that share the same THREADID belong to a single email thread. Before doing any computation, since the DATE field in email headers is in the local time zone, we converted all dates to UTC time so that we can measure reply times in threads whose participants live in different time zones. We also decided to ignore threads that have more than one recipient, since it is harder to argue that the first email in the thread was a question that needed a reply from all the recipients. Additionally, we only took the time difference between the first and second email in a given thread ignoring later emails from the same thread since it is harder to know if those emails are replies to a previous request. Figure 23 shows the median, 95 percentile and the 99 percentile reply time for every month between January 2006 and June 2013. The periodical peaks are due to seasonality trends with peaks around the New Year holidays (December and January) and summer vacation (July).

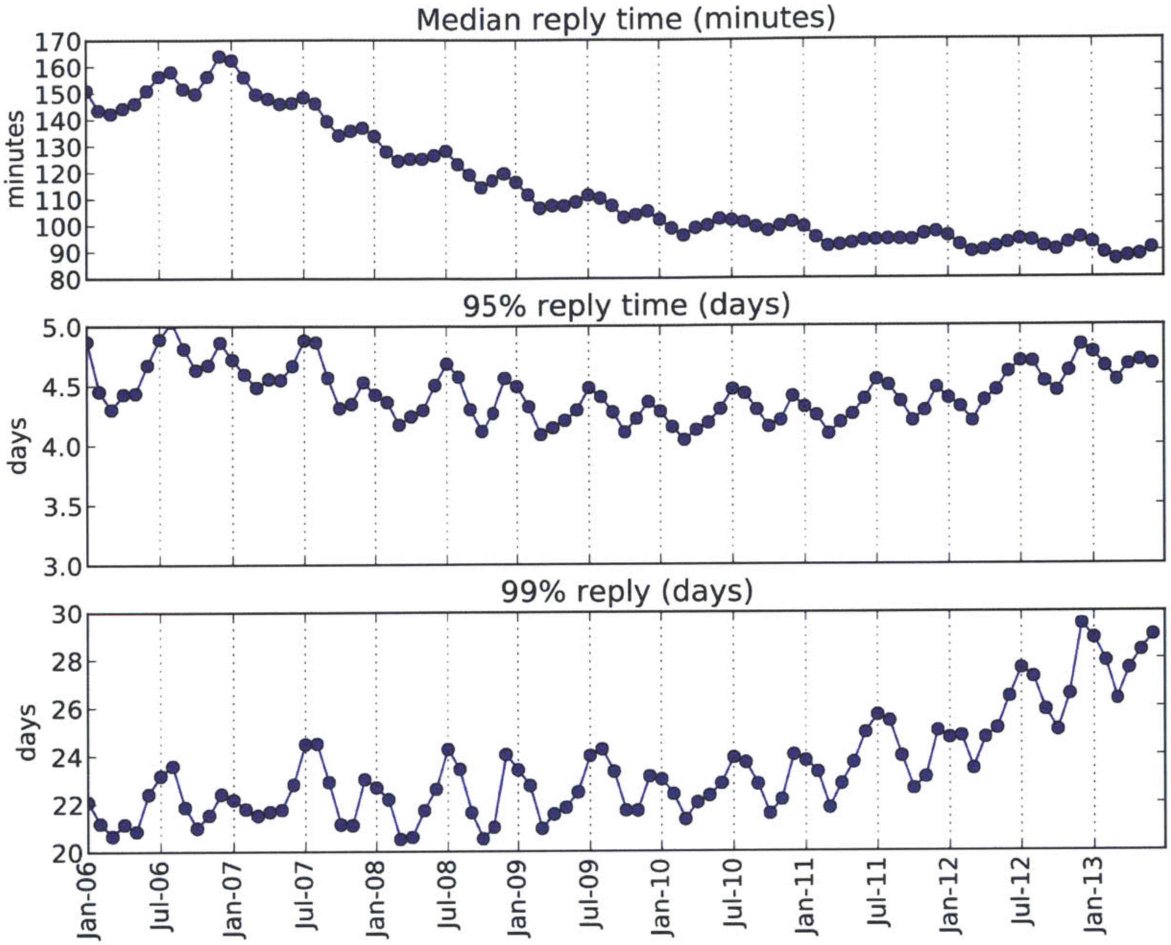


Figure 23 Reply times trend from 2006 to 2013

The median reply time has improved significantly from approximately 160 minutes in January 2007 to 90 minutes in June 2013, which is a reduction of 44%. This is most likely caused by increased Internet availability, online communication, and smart phones, making it easy for us to read and reply to emails more often. The 95 percentile reply time however hasn't changed significantly in the past 7 years averaging around 4.5 to 5 days. The 99 percentile on the other hand has an upper trend, which may be a signal of the email overload problem. More concretely, it is increasingly harder to process lower priority emails in reasonable time because of the growing number of total received emails that we need to process.

While there is no question that the overwhelming number of emails impacts our free time, we have shown it has no effect on our performance to reply to emails. On the contrary, for the majority of emails we received, we are now almost twice as efficient in replying than 8 years ago.

Number of recipients and email usage on weekends

Figure 24a shows the yearly change in the average number of recipients per email from 2005 to 2013 along with the 95% confidence interval denoted by red error bars. While the differences between two consecutive years are not statistically significant, the difference between 2007 and 2013 is, pointing to the fact that we are slowly getting more collaborative.

Figure 24b shows the fraction of emails sent on a weekend along with the 95% confidence interval (red error bars). You can see a statistically significant increase in the fraction of emails that are sent on a weekend, which points to the fact that in the context of sending emails, the gap between weekends and a week day is decreasing. I hope this result catches our attention and sparks some thoughts and discussions about work-life balance. That being said, we are still far from treating a weekday and a weekend equally (in that case we would expect a fraction of $2/7 = 0.286$).

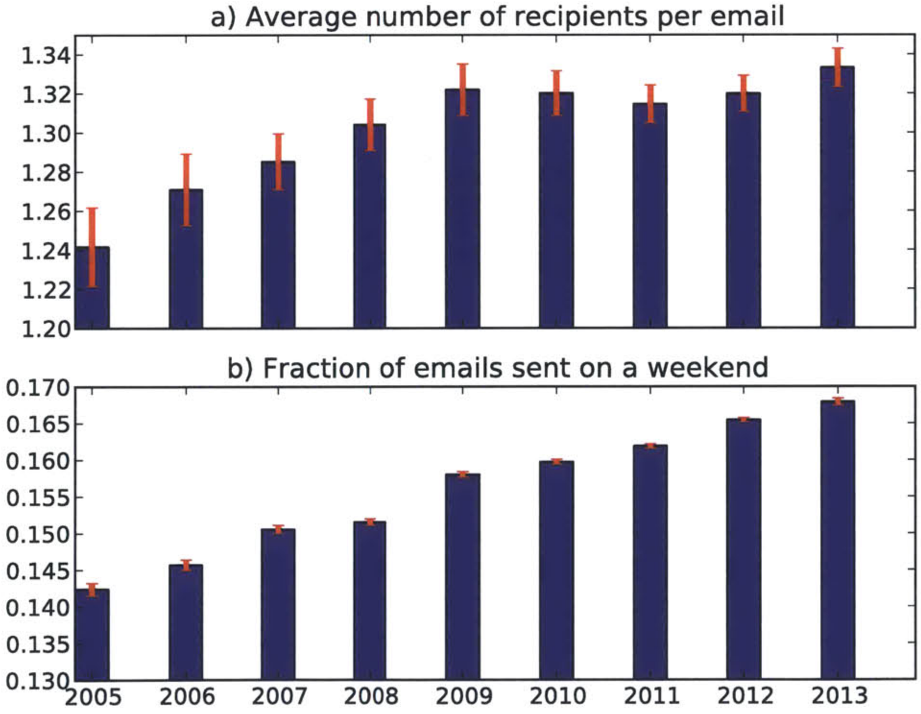


Figure 24 Email trends in the last 7 years along with the 95% confidence intervals denoted by a red line. a) Average number of recipients per email. b) Fraction of emails sent on a weekend.

5. CONCLUSION

We are overwhelmed by data, not in the sense that we don't have the resources and power to process it, but because we are in the early age of the information revolution and we are still learning to separate signal from noise. It will take some time to learn to see these signals, just like it took some time for evolution to build, mold and shape our senses.

But we are constantly making progress. Visualizations leave the world of traditional graphic design, replacing ink by pixels. New interactive data engines allow us to combine the best of both worlds - the blazingly fast speed of computers with the human intuition and intelligence. They provide a two-way street between the data and the human, turning the reader from a spectator to an explorer (Hidalgo, 2014). My hope is to be part of this revolution by making tools, and iterate on them as I learn to recognize signal from noise.

In this thesis, I focused on online communication because it is becoming more and more prominent in our everyday life. I focused on email in particular, because unlike most old technologies, it is simply too important and ubiquitous to be replaced. I presented Immersion, a tool build for the purposes to analyze and visualize the information hidden behind the digital traces of email activity, to help us reflect on our actions, learn something new about ourselves, quantify it, and hopefully make us react and change our behavior. I also touched on the email overload problem and work-life balance by quantifying general email usage using a large real-world email dataset.

Hopefully future work on Immersion will keep the site alive with new features revealing interesting new patterns, and the data analysis on the large dataset will continue and help us reason about the new information age where we are exposed to data that is much bigger than any human has directly experienced in the history of humankind.

6. BIBLIOGRAPHY

Albert, R., & Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of modern physics , 74* (1).

Bastian, M., Heymann, S., & Jacomy, M. (2009). Gephi: an open source software for exploring and manipulating networks. *ICWSM*, (pp. 361-362).

Batagelj, V., & Mrvar, A. (1998). Pajek-program for large network analysis. *Connections , 21* (2), 47-57.

Bayardo, R. J., & Agrawal, R. (2005). Data privacy through optimal k-anonymization. *21st International Conference on Data Engineering* (pp. 217-228). IEEE.

Bostock, M., Ogievetsky, V., & Heer, J. (2011). D^3 data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on , 17* (12), 2301-2309.

Clauset, A., Newman, M. E., & Moore, C. (2004). Finding community structure in very large networks. *Physical Review E , 69*.

Correa, C. D., & Ma, K.-L. (2011). Visualizing social networks. In C. C. Aggarwal, *An introduction to social network data analytics* (pp. 307-326). USA: Springer.

Crispin, M. (2003). *RFC 3501, Internet message access protocol, version 4, revision 1.* University of Washington. The Internet Society.

Cui, W. (2010). Context preserving dynamic word cloud visualization. *Pacific Visualization Symposium* (pp. 121-128). IEEE.

de Montjoye, Y.-A., Hidalgo, C. A., Verleysen, M., & Blondel, V. D. (2013). Unique in the Crowd: The privacy bounds of human mobility. *Scientific reports , 3*.

de Montjoye, Y.-A., Wang, S. S., & Pentland, A. (2012). On the Trusted Use of Large-Scale Personal Data. *Data Engineering Bulletin* , *35* (4), 5-8.

Duch, J., & Arenas, A. (2005). Community detection in complex networks using extremal optimization. *Physical review E* , *72* (2).

Eagle, N., & Pentland, A. (2006). Reality mining: sensing complex social systems. *Personal and ubiquitous computing* , *10* (4), 255-268.

Easley, D., & Kleinberg, J. (2010). *Networks, crowds, and markets* (Vol. 6.1). Cambridge Univ Press.

Fisher, D., Brush, A., Gleave, E., & Smith, M. (2006). Revisiting Whittaker & Sidner's email overload ten years later. *In Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work* (pp. 309-312). ACM.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports* , *486* (3), 75-174.

Frau, S., Roberts, J. C., & Boukhelifa, N. (2005). Dynamic coordinated email visualization. *13th International Conference on Computer Graphics, Visualization and Computer Vision* (pp. 187-193). UNION Agency.

Goldstein, J. (2013, July 1). *An MIT Project That Lets You Spy On Yourself.* Retrieved from NPR: Planet Money: http://www.npr.org/blogs/money/2013/07/01/197632066/an-mit-project-that-lets-you-spy-on-yourself

Han, J., Kamber, M., & Pei, J. (2006). *Data mining: concepts and techniques.* Morgan kaufmann.

Hassan-Montero, Y., & Herrero-Solana, V. (2006). Improving tag-clouds as visual information retrieval interfaces. *International Conference on Multidisciplinary Information Sciences and Technologies,* (pp. 25-28).

Hidalgo, C. A. (2014, March 17). *The Data-Visualization Revolution.* Retrieved from Scientific American: http://www.scientificamerican.com/article/the-data-visualization-revolution/

LinkedIn. (n.d.). Retrieved April 15, 2014, from InMaps: http://inmaps.linkedinlabs.com/

MentionMapp. (n.d.). Retrieved April 15, 2014, from http://mentionmapp.com/

Microsoft Corporation. (2011). *EWS Java API 1.2.* Retrieved May 5, 2014, from http://archive.msdn.microsoft.com/ewsjavaapi/Release/ProjectReleases.aspx?ReleaseId=5754

Moreno, J. L. (1946). Sociogram and sociomatrix. *Sociometry , 9,* 348-349.

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences , 103* (23), 8577-8582.

Newman, M. E. (2003). The structure and function of complex networks. *SIAM review , 45* (2), 167-256.

Newman, M. E., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E , 69* (2).

O'Madadhain, J. (2003). *The jung (java universal network/graph) framework.* University of California, Irvine, Irvine, CA.

Pentland, A. (2009). *Reality mining of mobile communications: Toward a new deal on data.* World Economic Forum.

Perer, A., & Smith, M. A. (2006). Contrasting portraits of email practices: visual approaches to reflection and analysis. *Proceedings of the working conference on Advanced visual interfaces* (pp. 389-395). ACM.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program: electronic library and information systems , 14* (3), 130-137.

Quantified Self, Wikipedia. (n.d.). Retrieved April 15, 2014, from http://en.wikipedia.org/wiki/Quantified_Self

Rahm, E., & Do, H. H. (2000). Data cleaning: Problems and current approaches. *Data Eng. Bull. , 23* (4), 3-13.

Riesman, A. (2013, June 30). *What your metadata says about you.* Retrieved from Boston Globe: http://www.bostonglobe.com/ideas/2013/06/29/what-your-metadata-says-about-you/SZbsH6c8tiKtdCxTdl5TWM/story.html

Rosvall, M., & Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences , 105,* 1118-1123.

Schneier, B. (2013). *The Battle for Power on the Internet.* Retrieved April 15, 2014, from Bruce Schneier blog: https://www.schneier.com/blog/archives/2013/10/the_battle_for_1.html

Smilkov, D., Jagdish, D., & Hidalgo, C. A. (2013). Retrieved from Immersion: http://immersion.media.mit.edu

Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems , 10* (05), 557-570.

The Radicati Group. (2012). *Email market 2012-2016.* Retrieved April 29, 2014, from Radicati: http://www.radicati.com/wp/wp-content/uploads/2012/10/Email-Market-2012-2016-Executive-Summary.pdf

The Radicati Group. (2013). *Email market 2013-2017.* Retrieved April 29, 2014, from Radicati: http://www.radicati.com/wp/wp-content/uploads/2013/11/Email-Market-2013-2017-Executive-Summary.pdf

TouchGraph. (n.d.). Retrieved April 15, 2014, from TouchGraph: http://www.touchgraph.com/navigator

Viégas, F. B., Golder, S., & Donath, J. (2006). Visualizing email content: portraying relationships from conversational histories. *Proceedings of the SIGCHI conference on Human Factors in computing systems* (pp. 979-988). ACM.

Viegas, F. B., Wattenberg, M., & Feinberg, J. (2009). Participatory visualization with wordle. *Visualization and Computer Graphics, IEEE Transactions on , 15* (6), 1137-1144.

Viegas, F. (2005). *Mountain.* Retrieved April 20, 2014, from Mountain: http://alumni.media.mit.edu/~fviegas/projects/mountain/index.htm

Ward, M., Grinstein, G., & Keim, D. (2010). *Interactive data visualization: foundations, techniques, and applications.* Natick, Massachusetts, USA: AK Peters, Ltd.

Wasserman, S. (1994). *Social network analysis: Methods and applications* (Vol. 8). Cambridge univ press.

Whittaker, S., & Sidner, C. (1996). Email overload: exploring personal information management of email. *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 276-283). ACM.

Wolfram Alpha. (n.d.). Retrieved April 15, 2014, from Wolfram|Alpha Personal Analytics for Facebook: http://www.wolframalpha.com/facebook/

Wolfram, S. (2012). *The personal analytics of my life*. Retrieved April 15, 2014, from Stephen Wolfram Blog: http://blog.stephenwolfram.com/2012/03/the-personal-analytics-of-my-life/

World Economic Forum. (2011). *Personal Data: The Emergence of a New Asset Class.* World Economic Forum.