

POLITECNICO DI MILANO
Scuola di Ingegneria Industriale e dell'Informazione
Corso di Laurea triennale in Ingegneria Fisica

Pattern ricorrenti nella comunicazione umana scritta



Relatore: Prof. Paolo Biscari

Tesi di Laurea triennale di:
Giacomo Bassi - matr. 912153
Michelangelo Olmo Nogara Notarianni - matr. 917451

19 settembre 2021

Indice

0.1	Introduzione	3
1	Leggio	5
1.1	Caratteristiche dell'analisi di big data	5
1.2	Wolfram Mathematica ^(R)	6
1.3	Il dato: le e-mail	7
1.4	Il protocollo Postfix	9
1.5	Diagramma di flusso	11
2	Pulizia dei dati	13
2.1	Prima pulizia: e-mail consegnate correttamente	13
2.2	<i>datiOrdinati</i> e gli alias	14
2.3	Grafo di una settimana di e-mail	16
3	Elaborazione dei dati	19
3.1	Utenti	19
3.2	Iistogrammi	22
3.3	Selezione utenti	24
3.4	Codice per sei mesi - <i>secondoDatiOrdinati</i>	25
4	Ricerca di leggi	27
4.1	Il tempo	27
4.2	Gestione delle risposte	31
4.3	Nuvole	32
4.4	Ventagli	36
4.5	Importanza percepita, comportamento sociale	39
5	Conclusioni	42

0.1 Introduzione

Questa tesi mira a identificare delle ricorrenze nella comunicazione umana scritta nell'ambito dei big data. I metodi utilizzati sono quelli propri della statistica applicata a un vasto database. Nel nostro caso questo consiste in qualche decina di milioni di righe, insignificanti se prese singolarmente per via della loro varietà. Perciò gran parte del lavoro si è sviluppato attorno ad alcune principali domande. Come rappresentare efficacemente un grande volume di dati? Quali elaborazioni sono efficienti, in termini di informazioni ottenute, considerato il tempo e la potenza di calcolo necessari? Quali e quante imprecisioni si possono introdurre in modo da aumentare tale efficienza? Il tema che proponiamo è lo scambio di posta elettronica in cui è coinvolta una grossa organizzazione, per esempio un'università, a noi sconosciuta. Premettiamo che simili studi si possono fare, e sono già stati fatti, per altre forme di comunicazione scritta come le lettere cartacee o la messaggistica istantanea. Si veda per esempio *Hidden scaling patterns and universality in written communication* [1]. È stupefacente scoprire che, nonostante le differenze ovvie, vi sono molti fenomeni qualitativamente e quantitativamente analoghi anche fra i diversi canali comunicativi. Il campione da noi analizzato non è, come detto, formato da comunicazioni tra persone scelte casualmente. Le e-mail presenti sono scambiate da utenti interni a questa organizzazione durante l'anno accademico 2016/2017 e comprendono sia comunicazioni interne che da o verso l'esterno per un totale di 433500 indirizzi diversi solo nella prima settimana. Si capisce che si tratta di una grande rete: consideriamo per esempio di un'università. Così, per spiegare le differenze nelle attività di tutti gli utenti che compongono il nostro dato, ci potremo riferire ad un ipotetico rettore, il quale presumibilmente scambia molte più e-mail di un professore così come quest'ultimo ne invia e riceve molte più di una matricola. La caratterizzazione degli utenti sarà per certi versi un fulcro su cui il lavoro verterà. Il fatto che il nostro dato sia particolare comunque non deve scoraggiare, dal momento che parte del nostro studio è la ricerca di leggi “senza scala”: regole semplici che descrivono ugualmente la comunicazione di un solo utente e quella di un gruppo di persone fino ad arrivare, eventualmente, a rappresentare l'intera rete di corrispondenze a noi disponibile. Grazie ai metodi dell'analisi di Big Data, cercheremo dei pattern ricorrenti in questo campione molto complesso, che immaginiamo spaziare dagli intensi scambi da ufficio del lunedì mattina alla mail di conferma per l'acquisto di una bicicletta su eBay. Ripetiamo che questa complessità non è esente da regole semplici: per esempio, ci aspettiamo che le e-mail del lunedì mattina siano ben più numerose di quelle della domenica sera. Uno dei nostri obiettivi è mostrare se esiste o meno una correlazione tra il volume di posta scambiata

da un utente con i suoi diversi corrispondenti e il tempo medio di risposta a/da questi. Tali e altre regole accrescono la conoscenza del fenomeno e ci auguriamo siano utili sia per costruire canali di comunicazione più efficienti, regolati da protocolli sempre più universali, e sia per guidare ognuno di noi ad un uso più consapevole della propria casella di posta elettronica. Le nostre conclusioni contengono anche considerazioni estendibili ad altri campi: a partire dai metodi di analisi e rappresentazione che abbiamo dovuto definire, per esempio, fin dal primo capitolo, nell'ambito della programmazione fino allo sfociare nella sociologia e nella psicologia comportamentale. Grazie al coordinamento del Professor P. Biscari, la tesi è inserita in un progetto pluriennale portato avanti ogni anno da laureandi triennali del corso di Ingegneria Fisica del Politecnico di Milano. La base di questo progetto è stata la tesi magistrale sviluppata da Massimiliano Piccini nel 2015, accompagnata dalla tesi del 2020 di Barboni, Carella, Notarangelo. Queste fonti, di cui spesso riporteremo le considerazioni con eventuali commenti, sono riportate in bibliografia.

Capitolo 1

Leggìo

1.1 Caratteristiche dell’analisi di big data

In generale, quando parliamo di *big data* ci riferiamo alla grande estensione di un insieme di dati, ma anche alla sua eterogeneità e alla scarsa rilevanza del dato grezzo preso singolarmente. Per registrare *megadati* è necessaria una misurazione molto veloce, che tipicamente viene fatta automaticamente da software o hardware dedicati. Nel nostro caso, il risultato di questa “misurazione” è una riga di caratteri all’interno di un file di centinaia di megabyte, cioè una piccola goccia nella più grande delle bottiglie. Si capisce che conoscere una sola di queste righe è inutile: sapere che cosa stia facendo il mail server in una data frazione di secondo è insignificante. La logica dell’analisi nell’ambito dei big data permette di leggere queste righe con algoritmi suggeriti dalle caratteristiche del database stesso e fa emergere informazioni di carattere diverso rispetto a quelle ottenute dalla singola riga. Inoltre, data la velocità con cui si registrano dati disomogenei per una comunità molto numerosa, è necessario che i processi software che gestiscono ed elaborano questi dati siano altrettanto veloci. Se i tempi di gestione fossero troppo lunghi rispetto a quelli con cui potenzialmente si ottengono i dati, questo genere di analisi sarebbe obsoleta in partenza: nel 2017 sono state inviate, mediamente nel mondo, 2.4 milioni di email al secondo; nel 2019, 2.7 milioni; a fine agosto del 2021 le email inviate sono in media più di 3 milioni al secondo [si veda p.e. internetlivestats.com]. Sebbene il nostro studio si riferisca a delle comunicazioni e-mail di cinque anni fa, abbiamo scritto dei programmi che, per quanto riguarda l’analisi della nostra rete di utenti, funzionerebbero in tempo reale anche su comuni calcolatori portatili. Non solo abbiamo speso molto tempo e diversi tentativi per ottimizzare il nostro codice dalla lettura delle e-mail (dato grezzo) allo studio del comportamen-

to di alcuni utenti “eletti”, ma abbiamo anche suddiviso lo svolgimento di questi processi in diversi sottoprogrammi in modo che si potessero svolgere facilmente in parallelo su diversi calcolatori. Nel nostro caso dovremo fare un’analisi descrittiva delle comunicazioni di posta elettronica registrate da tre mail server; molti dei dati raccolti sono poco interessanti (per esempio messaggi tecnici) o comunque non sono nel formato richiesto. Tra tutti i nostri dati ci sono inoltre diversi errori, come per esempio una larga parte di e-mail non correttamente consegnate al destinatario. È importante che il primo passo sia evidenziare le informazioni a noi utili, prelevarle dal dato originario, “pulirle” da eventuali errori e rappresentarle opportunamente per le analisi che ci aspettiamo di fare. Per esempio, la prima rappresentazione che otteniamo dal dato raggruppa tutte le righe relative ad una stessa e-mail in un’unica struttura e ne trasforma la data e l’ora in un valore intero che conta i secondi passati dall’inizio dell’anno 2016. In questo modo potremo facilmente costruire una macrostruttura “utente” che descriva tutte le comunicazioni di un dato account – che, come vedremo, spesso non coincide con un solo indirizzo email - interno alla nostra rete.

1.2 Wolfram Mathematica®

Per questo ci siamo serviti del software *Wolfram Mathematica*®. Tale applicazione permette di scrivere espressioni matematiche simboliche nel linguaggio nativo “Wolfram Language” ed elaborarle grazie all’ambiente computazionale integrato. L’elaborazione avviene con ottime prestazioni in termini di tempo e affidabilità: più volte ci è capitato di eseguire programmi diversi con la stessa funzione per capire quale fosse il più prestante, misurandone il tempo computazionale data l’aderenza dei risultati alle aspettative. Questi test hanno evidenziato come le funzioni native di Mathematica fossero da preferire, cosa che ci ha portati a imparare uno stile di programmazione in un primo momento meno intuitivo, ma più aderente alle possibilità offerte dal linguaggio. Un esempio importante sono le associazioni, indicate come $\langle|\text{Key}\rangle\text{Value}|\rangle$, una struttura dati tipica di Wolfram Language. Rispetto alle più intuitive matrici, le associazioni permettono di accedere ai valori senza contatore e in maniera diretta tramite la key, che è a sua volta un’espressione arbitrariamente scelta.

Certamente è difficile rendere chiaro un algoritmo in un linguaggio nuovo e molto simbolico. Abbiamo cercato di rispettare uno standard di leggibilità e stabilità per un secondo utilizzatore, dividendo il lavoro fra diversi sottoprogrammi cosicché svolgessero funzioni precise e che fossero facilmente “chiamabili” e adatte a essere inserite in un circuito più ampio. Oltre

```
tabAlias = AssociationThread[
  alias[[All, 1]] → alias[[All, 2]]];
(*associa ad ogni utente (KEY) la lista degli
alias relativi (VALUES)*)|
```

Figura 1.1: Esempio di dichiarazione di associazione

```
If[! MissingQ[tabAlias[u]],
  nAlias = Length[tabAlias[u]],
  nAlias = 0];
uRicT = ricA[u]; (*mail ricevute*)
uRicP = ricPA[u];
(*mail ricevute pesate con nrcpt*)
```

Figura 1.2: Accesso al valore di un’associazione

al vantaggio, già discusso, di avere la possibilità di svolgere calcoli diversi in parallelo, con la suddivisione in sottoprogrammi e inserendo all’interno del codice interi paragrafi di commenti esplicativi (Fig. 1.3) abbiamo reso il nostro lavoro anche più comodamente correggibile. Ci piace infatti pensare che questa tesi sia in continuità con i lavori dei nostri colleghi negli anni precedenti e negli anni a venire.

1.3 Il dato: le e-mail

Il dato di partenza è tecnicamente un file di tipo .log, composto da decine di milioni di righe che registrano l’attività di tre diversi mail server nell’arco temporale di circa un anno a cavallo del biennio 2016/17. In figura 1.4 riportiamo una ventina di queste righe.

Si nota che ogni riga inizia dalla data e dall’ora “correnti” in cui il messaggio è stato registrato nel file dai tre mail server, ai quali sono associati i nomi *bronze* (visibile in fig. 1.4), *silver*, *gold*. Come discusso nella tesi di Barboni, Carella, Notarangelo [2], il fatto che i mail server siano tre differenti non complica il nostro dato, in quanto gli indirizzi e-mail sono mascherati nello

```

datiDaSalvare = {}; (* qui scrivo i dati che poi verranno salvati nel file
"dati ordinati" *)
tempMails = {};
(* in tempMails lascio "in coda" tutte le email non ancora
completate: quelle che non hanno ancora tanti "to" quanti detti in nrcpt,
o quelle il cui status non è "expired",
conservandole anche tra un file log e il successivo *)
i = 1;

While[i ≤ Length[righe], (*si leggono tutte le righe, una per volta*)
  If[i ≠ 1 && StringTake[righe[[i - 1]], {1, 3}] == "Dec" &&
    StringTake[righe[[i]], {1, 3}] == "Jan", anno++];
  (* l'anno aumenta al passaggio tra Dec e Jan *)
  (* salvo solo le righe che contengono una delle stringhe qui sotto
  (smtp, lmtp, qmgr,...) *)
]

```

Figura 1.3: Frammento del programma *datiOrdinati*

```

Oct 9 06:29:41 bronze postfix/cleanup[2573]: D05D81119F: message-id=<809c839fcb5ed50927181db13ba3b6ca>
Oct 9 06:29:41 bronze postfix/postfix[13771]: D05D81119F: from=<83c4ff419172ddb6ed3288d8e09c7eb>, size=11356,
Oct 9 06:29:42 bronze postfix/lmtp[2178]: D05D81119F: to=<db03e830a56eeeba03d875615b476fb>, relay=85c1228
Oct 9 06:29:48 bronze postfix/cleanup[2573]: 133541119F: message-id=<ca86d3424fcfeacea2795d1e3f51641>
Oct 9 06:29:48 bronze postfix/qmgr[13771]: 133541119F: from=<5d9eedcc557bddd7d0369ba1ef6a012b>, size=49963,
Oct 9 06:29:48 bronze postfix/postfix[2574]: 133541119F: to=<192b134b1f6d40c7b8239f7561bee1f2>, relay=49b85b0
Oct 9 06:29:49 bronze postfix/cleanup[2573]: 3E3391119F: message-id=<600fa399bb503e2f6a6c3629c7283623>
Oct 9 06:29:49 bronze postfix/qmgr[13771]: 3E3391119F: from=<cf949042d0c8e35657c0000d2c884957>, size=13026,
Oct 9 06:29:49 bronze postfix/postfix[2392]: 3E3391119F: to=<45dbe63a8cb60f8d790beff8f071d93>, relay=85c1228
Oct 9 06:29:49 bronze postfix/smtp[32386]: 3E3391119F: to=<57c68bf8545a4428cad4e86401d0d9d1>, orig_to=<45db
Oct 9 06:29:51 bronze postfix/cleanup[2573]: 16F0C1119F: message-id=<98f9e0adc4ff381270a22dde356ac294>
Oct 9 06:29:51 bronze postfix/qmgr[13771]: 16F0C1119F: from=<886e6127e55bcf49cd076abcbd192540>, size=11365,
Oct 9 06:29:51 bronze postfix/lmtp[2574]: 16F0C1119F: to=<b5d140cb5754edb6a6ec25904a59e73>, relay=cdb07fc
Oct 9 06:29:52 bronze postfix/cleanup[2573]: 1BACE1119F: message-id=<e083bf73b462a84b2a85b90853d576f3>
Oct 9 06:29:52 bronze postfix/qmgr[13771]: 1BACE1119F: from=<ed9261272d8bb318984baa32b7aadd1b>, size=12239,
Oct 9 06:29:52 bronze postfix/lmtp[2178]: 1BACE1119F: to=<a70a39c72c10e83f61b45f9ada84fa5>, relay=cdb07fc
Oct 9 06:29:53 bronze postfix/cleanup[2573]: 0AE0D1119F: message-id=<333549f733edfbdd54d42527b30c4924>
Oct 9 06:29:53 bronze postfix/qmgr[13771]: 0AE0D1119F: from=<54a9e6e20948ebe96d8da6c139a37d0>, size=16881,
Oct 9 06:29:53 bronze postfix/lmtp[2574]: 0AE0D1119F: to=<878bf20b526642535835a6a88fa208e6>, relay=cdb07fc
Oct 9 06:30:15 bronze postfix/master[11580]: reload -- version 2.9.6, configuration /etc/postfix
Oct 9 06:30:15 bronze postfix/master[11580]: warning: ignoring inet protocols parameter value change

```

Figura 1.4: Porzione del file log utilizzato

stesso modo da tutti e tre¹. La stringa contiene poi un messaggio di tipo *postfix* che identifica il protocollo secondo il quale il mail server agisce. Nei diversi momenti di gestione della e-mail in questione, identificata dal codice maiuscolo di dieci cifre (in base esadecimale) che segue, viene chiamato un comando (in gergo, un demone) per una specifica operazione; approfondiremo la questione nel prossimo paragrafo. Consideriamo per esempio la email "3E3391119F": a questa corrispondono, dalla settima alla decima riga, un

¹Quanto detto è suggerito dal fatto che gli stessi indirizzi in codice esadecimale compaiono nei *log* dei diversi mail server; si ha una conferma ulteriore nel considerare che questi indirizzi mantengono anche la stessa cerchia di contatti.

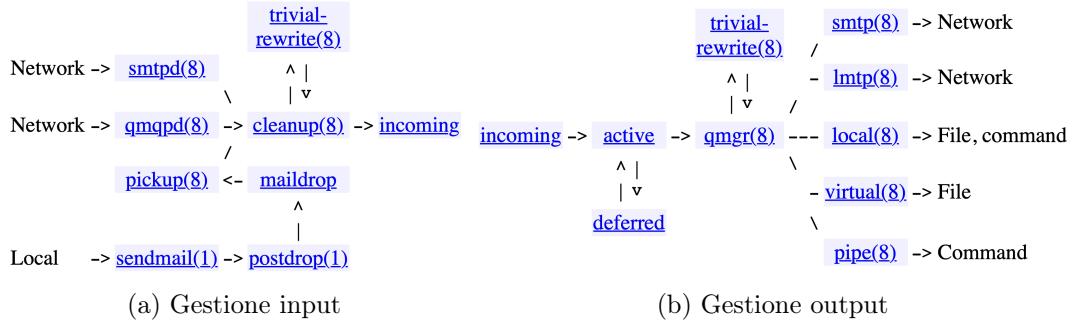
message-id, un mittente nella riga di *from* e due destinatari nelle righe di *to*. Gli indirizzi e-mail sono mascherati in codice esadecimale: della e-mail in esempio il mittente è “cf949042d0c8e35657c0000d2c884957”, riconoscibile dal nostro programma in quanto compreso fra i simboli “<” e “>”. In generale una e-mail è intestata insieme al suo peso in byte e al numero di destinatari (*nrcpt*), che però non sempre corrisponde al numero effettivo di questi². Il resto dei caratteri che compongono il file *.log*, per esempio le due righe in fondo alla figura 1.4, sono messaggi tecnici legati al mail server e non sono direttamente utili ai fini della nostra ricerca. Dedichiamo un paragrafo a Postfix, il più usato fra i mail server, in quanto parte integrante della preparazione del nostro database.

1.4 Il protocollo Postfix

I mail server (“Mail Transfer Agent” (MTA), in italiano *Demoni di Posta Elettronica*) sono sistemi di gestione di e-mail che funzionano sulla base di protocolli. Dalla fine degli anni ’90, l’MTA più comunemente utilizzato è Postfix, poiché rispetto al precedente *Sendmail* offre maggiori garanzie di sicurezza e stabilità in caso di *buffer overflow*. La sigla “postfix” all’inizio di ogni riga nella figura 1.4 indica che il server da cui provengono i dati opera grazie a Postfix. Per capire il ruolo che un mail server ha all’interno di una comunicazione si pensi per semplicità ad una tradizionale lettera postale mandata da A a B. Perché la lettera venga consegnata è necessario un sistema postale a più livelli fatto di uffici centrali e centri di smistamento locali che si interfacciano con i cittadini. Il mail server, e in particolare Postfix, ha la funzione di centro di smistamento locale, e quindi è caratterizzato da una serie di indirizzi di competenza interni e da una rete a cui affacciarsi in caso di comunicazioni verso l’esterno. Postfix deve possedere due interfacce, una di ingresso, detta *inbound interface*, e una di uscita, *outbound interface*. Ogni interfaccia ha diverse porte di ingresso e di uscita: queste sono usate opportunamente a seconda del contenuto della comunicazione (file, comando, etc.) e a seconda del tipo di canale aperto. I canali possono essere aperti verso altri *mail processing systems* o verso terminali, che a loro volta possono essere interni, identificati come *local*, o esterni, identificati come *network*, rispetto alla rete gestita. In figura 1.5 riportiamo gli schemi delle due interfacce.

I *daemons*, o comandi, che interessano quasi tutto il nostro dataset sono i seguenti:

²Per esempio, quando sbagliamo a scrivere l’indirizzo del destinatario o quando le e-mail sono inviate in blocco ad indirizzi pseudocasuali, il numero di destinatari effettivo è minore di *nrcpt*

Figura 1.5: Architettura di Postfix (da *postfix.org*)

cleanup: il messaggio è entrato nel server dall'inbound interface, quindi è preso in carico da Postfix tramite il comando *cleanup*, che lo aggiunge all'incoming queue;

qmgr: il centro funzionale del server Postfix. QueueManager distribuisce le consegne alle porte di uscita delegando la spedizione ai demoni: *local*, *smtp*, *lsmtp*, *pipe*. Inoltre organizza i pacchetti mettendo insieme file, indirizzo mittente e uno o più indirizzi destinatari, generando una *active queue* o una *deferred queue* nel caso in cui Postfix non possa inviare immediatamente il messaggio;

lsmtp: spedisce i pacchetti ad altri mailbox servers locali o remoti, con cui comunica secondo il *Local Mail Delivery Protocol*;

bounce: si occupa di notificare al mittente un invio non riuscito. Tale evenienza è gestita da *qmgr* che chiama *bounce* in caso di necessità;

smtp: spedisce i messaggi in uscita a destinatari “lontani”, provando ogni indirizzo di una lista di *mail exchangers* finché non ha risposta da uno dei server;

local: si occupa di consegne nelle mailboxes di indirizzi locali avendo a disposizione i dati relativi agli utenti;

pipe: è l'interfaccia di uscita per altri protocolli. Il comando chiama altri e diversi programmi per regolare il corpo del file secondo i propri standard.

Dallo studio dell'architettura di Postfix evinciamo un'utile conclusione: per ogni comunicazione andata a buon fine esiste una catena ordinata di comandi. In particolare ci aspettiamo di trovare nella prima riga relativa ad un nuovo

messaggio il comando *cleanup*, seguito eventualmente dal comando *qmgr* e poi da uno dei comandi di uscita (*lsmtp, smtp, local, pipe*). Con questa consapevolezza abbiamo progettato il programma per la prima pulizia dei dati, eliminando le righe che non fossero intestate con uno di questi sei comandi, ottenendo enormi vantaggi in termini di tempo computazionale. In figura 1.6 è riportato un particolare del codice in cui è evidente tale passaggio: se non c'è scritto nulla di buono, passa alla riga successiva.

```
(* salvo solo le righe che contengono una delle stringhe qui sotto (smtp, lsmtp, qmgr,...) *)
If[StringFreeQ[righa[[i]], {"/cleanup" ~~ __ ~~ "}, "/local" ~~ __ ~~ "], "/smtp" ~~ __ ~~ "], "/qmgr" ~~ __ ~~ "],
"/lsmtp" ~~ __ ~~ "], "/pipe" ~~ __ ~~ "]], ,
```

Figura 1.6: Frammento del programma *datiOrdinati*

1.5 Diagramma di flusso

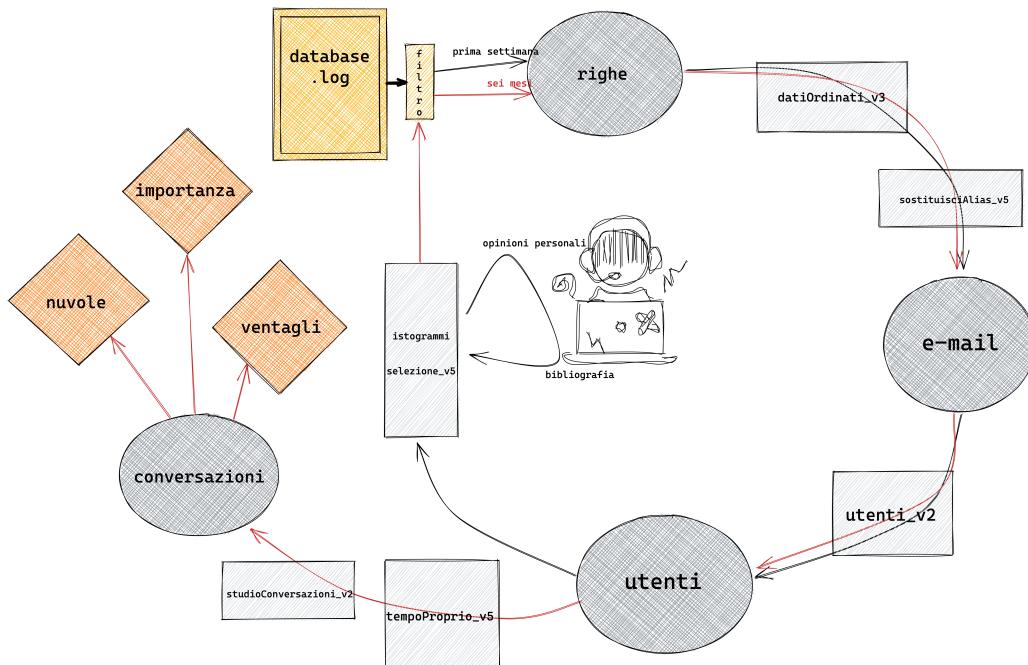


Figura 1.7: Schema del percorso seguito dal dato grezzo agli studi conclusivi. Realizzato grazie allo *sketching tool* offerto da *excalidraw.com*

In figura 1.7 riportiamo il percorso che, a grandi linee, porta dal dato grezzo iniziale del file *log* (rappresentato dal blocco giallo in alto a sinistra) alle

nostre conclusioni (in arancione). Ognuna delle cornici tondeggianti esplicita la rappresentazione³ corrente del dato, elaborato nell'una e nell'altra forma grazie a tutti i programmi riportati come rettangoli in grigio chiaro. Idealmente il dato grezzo potrebbe fare l'intero percorso, tuttavia i tempi computazionali sono troppo lunghi per elaborare tutto il database, ricco di tanti dati interessanti quanti sono quelli inutili per arrivare alla fine. Decidiamo di analizzare solo la prima settimana di raccolta dei dati per cercare pochi utenti, ma buoni, che riescano a seguire il percorso della freccia nera in figura 1.7 fino a superare la selezione. Pensando che ci sarebbero state sufficienti le informazioni complete su cinque buoni utenti per discutere prove e controprove, ne abbiamo scelti dieci poiché il comportamento valutato su una settimana può non essere indicativo del comportamento reale. Per questi pochi utenti, selezionare il dato utile da portare fino allo studio esplicito delle conversazioni di tutto un semestre richiede meno "sforzi" e restituisce molti più risultati rispetto alla visione completa di tutte le e-mail registrate in una settimana.

³Il dato di partenza, come detto, è un file *log*. Osservato direttamente sullo schermo mostra un sacco di cose, ma non sembrano esserci particolari pattern ricorrenti della comunicazione umana scritta. Grazie ai programmi/operatori in figura 1.7, trasformiamo delle righe di testo in e-mail, poi in schede utente e infine in vere e proprie conversazioni di un utente con tutti i suoi corrispondenti. In questo senso ci piace parlare di *rappresentazioni*.

Capitolo 2

Pulizia dei dati

2.1 Prima pulizia: e-mail consegnate correttamente

Avendo in mente il ruolo dei *demoni* Postfix, per prima cosa eliminiamo dal dato le righe che sono per intero inutili. Quello riportato in figura 1.6 è un passaggio importante e riduce molto i tempi di lavoro. Tuttavia gli ingredienti non sono ancora sufficienti per lanciarsi sulle conversazioni, infatti non tutte le righe che abbiamo salvato corrispondono ad una e-mail interessante. Come detto nel paragrafo sui *megadati*, data questa grande mole di informazioni, è necessario saper estrarre dal dato grezzo le componenti essenziali o quantomeno utili. Prima di tutto, potrebbero trovarsi errori di ogni genere: a noi interessa che un messaggio contenga un mittente e almeno un destinatario “effettivo” a cui la e-mail è stata davvero consegnata. Inoltre, è fondamentale avere un programma che legga, elabori e salvi queste informazioni in modo molto ordinato e senza accumulare errori o dubbi in quanto divergerebbero rapidamente. Il programma *datiOrdinati* inizia generando una nuova struttura dati ogni volta che legge una riga che caratterizza un mittente. Questa riga, nel file generato dal mail server, è sempre la prima -tra quelle che includiamo- a comparire fra le altre relative ad una data e-mail. In questa struttura, che potrebbe essere poi tenuta buona o no, si salvano il codice alfanumerico che caratterizza l'e-mail (*idmail*), la data e l'ora e il numero *nrcpt* di destinatari previsti, tutti elementi che dovrebbero essere presenti in una riga di tipo *from* registrata correttamente. La data e l'ora, per comodità, vengono convertite in secondi a partire dall'inizio del 2016 (cioè la mezzanotte del primo gennaio è il nostro tempo zero) e salvate come numero intero. A questo punto la scheda viene messa da parte e il programma continua a leggere il file -eventualmente generando altre schede-

finché non trova altre $n=nrcpt$ nuove righe che contengano lo stesso codice *idmail*: *nrcpt* è il numero di destinatari *previsti*, ma non tutte le email sono consegnate. Alcune hanno per status “deferred” o “bounced” poichè il mail server non conosce o non trova il destinatario, e dopo alcuni tentativi di invio si ritrova un’ultima riga che modifica lo status della mail in “expired”. Tutte queste ultime sono un’altra categoria di messaggi che scartiamo subito e il numero di destinatari sarà poi ricalcolato come numero di utenti a cui la e-mail è inviata correttamente, ovvero quelle il cui status associato è “sent”.

2.2 *datiOrdinati* e gli alias

La situazione è complicata dal fatto che un indirizzo di posta elettronica sia spesso direttamente collegato ad indirizzi differenti (nel nostro caso *nome.cognome@polimi* oppure *codicepersona@polimi* sono entrambe maschere per lo stesso account utente). Chiamiamo questi indirizzi usati da una stessa persona *alias* di un indirizzo principale. Se inviassi una email ad un alias, verrebbe consegnata correttamente allo stesso account che cerco, legato a un certo indirizzo principale; di più, da diversi servizi di posta elettronica è possibile inviare un messaggio usando come indirizzo un alias, cioè mostrarsi con diversi nomi. In quanto a noi non interessa tanto conoscere l’attività di un indirizzo, ma speriamo di avvicinarci alla completa attività di una persona (almeno per quanto riguarda un account legato alla nostra università-azienda), sarebbe opportuno conoscere tutti gli alias di ogni utente e ricollegare tutte le diverse e-mail all’indirizzo principale. Tuttavia, non sempre il mail server evidenzia un indirizzo come alias. Se l’alias comparisse tra i destinatari della mail e fosse riconosciuto dal mail server, comparirebbero due campi differenti nel file *log*: “orig_to” e “to”, come si intravede nella decima riga della figura 1.4. In questi casi *to* è il campo che contiene l’indirizzo principale e *orig_to* quello che contiene l’eventuale pseudonimo. In un primo momento, il nostro programma elimina tutti quei destinatari che compaiono con diversi alias nella stessa email, cosa che succede spesso nei messaggi inviati a numerosi utenti. Tuttavia questo procedimento non è sufficiente. Nel caso in cui l’account esaminato, interno o esterno alla nostra rete, fosse il mittente della mail, l’eventuale pseudonimo non verrebbe segnalato come tale: non esiste un campo *orig_from*. Risolveremo meglio la questione successivamente. A questo punto, il nostro programma si limita ad eliminare anche tutti quei destinatari che sono anche il mittente, cioè i messaggi autoinviai: anche nel caso in cui questi fossero “umani” -c’è chi si invia un promemoria o la lista della spesa- non li riterremmo interessanti in quanto non partecipi della comunicazione in quanto relazione sociale. Se il numero di righe lette

corrisponde a quello dei destinatari attesi e se dopo queste correzioni rimane almeno un destinatario effettivo⁴, la scheda viene archiviata, il programma se ne dimentica e continua la lettura del file. Con questo metodo riusciamo a salvare anche le email che compaiono a cavallo di diversi file *log*, in quanto rimangono archiviate nella loro scheda finché questa non è completa. Il risultato finale, che contiene tutte le email salvate, è una lista di liste di liste, cioè un vettore di un gran numero di strutture ognuna delle quali è composta da nove campi nel modo seguente:

1. Contatore intero;
2. Id-mail (codice alfanumerico che caratterizza la email);
3. Indirizzo del mittente;
4. Data e ora (in secondi) del messaggio di *from*;
5. Dimensione della email in KB;
6. Numero di destinatari effettivo;
7. Indirizzo del destinatario (eventualmente una lista di indirizzi);
8. Alias del destinatario (se presente, eventualmente una lista);
9. Data e ora (in secondi) del messaggio di *to* (eventualmente una lista).

Salviamo in due campi diversi i tempi dei messaggi di *from* e di *to* solo per qualche raro caso in cui il mail server ha troppe e-mail in coda e l'invio non riesce all'istante. La lista viene salvata da Mathematica in un file di testo, di cui riportiamo un frammento in figura 2.1.

Questa rappresentazione dei dati utili come struttura ci permette di fare considerazioni diverse in modo semplice. Per esempio, come mostreremo, disegnando un istogramma del numero di e-mail inviate nel tempo in una giornata si nota chiaramente un picco durante la mattinata, mentre su un anno si possono distinguere le vacanze natalizie ([2],[3]). Prima di andare oltre abbiamo cercato di ridurre i problemi con la presenza di pseudonimi tra gli indirizzi. Abbiamo scritto un programma che rilegge i nostri dati ordinati e genera una tabella con tutti gli indirizzi che sono certamente “principali” e il/i loro alias sfruttando le email in cui questi compaiono con i campi *orig_to* e *to*. Un secondo programma usa questa tabella per sostituire tutti

⁴Se le righe lette sono pari a *nrcpt*, ma per qualunque motivo nessuna delle e-mail è interessante, la scheda viene eliminata. Ritroviamo schede “a metà” solo per righe alle estremità tagliate dell’intero *log*

```

{1, "376BCA052", "9e156059f47de3d8b1b992e62dd72231", 26548234, 1266, 1,
 {"f8b26ca63c3f3ae740f15a54470fba98"}, {"7067fac259dbe3f6109460b0aff4ad95"}, {26548234}, {2, "16A2FA050", "ac3f61425e961072b3e8f18df13ad66b", 26548235, 2711, 1, {"72a662adcf91f75018f12271ffaa4203"}, {"ee126f6473adb73ca18df175926ab"}, {26548235}, {3, "0183AA050", "a534d31a693228daf031b852cd5a8ed6", 26548238, 11288, 2, {"6ea92ae894d05b58f6b3d10d9520db56", "d76cc95f5bc7b58061e6847174fc5d6"}, {"NoAlias", "6ea92ae894d05b58f6b3d10d9520db56"}, {26548238, 26548238}}, {4, "0D225A050", "3a696d8cbb296e9259643c84ba806957", 26548243, 21955, 1, {"1e1f8f6e6755f0ab470f50ba8e867ac"}, {"NoAlias"}, {26548243}}, {5, "CBE1DA052", "913f6c00e5e62a0f08216efe1df406d0", 26548247, 10627, 1, {"f61335d0dc2f18f2799161d8f226797a"}, {"NoAlias"}, {26548248}}, {6, "C2498A050", "5121d0bdbfaf2d3a66f0787b083f6868", 26548247, 12607, 2, {"3d39a5d8f66e52184b3d3fe2aec5fc2", "a721c0b7096fe51410e7ef46724d6fd1"}, {"NoAlias", "3d39a5d8f66e52184b3d3fe2aec5fc2"}, {26548247, 26548248}}, {7, "92122A050", "25519ecd074dff3cef971c16ce1334db", 26548248, 2718, 1, {"8ffd़da23491f18c3a839f9865fb7d4eb5"}, {"NoAlias"}, {26548248}}, {8, "6E774A050", "ebbca2f2376cc77b6f942c87d38a7109", 26548251, 10517, 1, {"fa6869796650549e29c6ff395a68eaa6"}, {"NoAlias"}, {26548251}}, 
```

Figura 2.1: Prime e-mail elaborate da *datiOrdinati*

gli pseudonimi che compaiono come mittenti (come detto, non direttamente riconoscibili) nella nostra lista. La correzione funziona: abbiamo prova del fatto che gli alias sono molto usati⁵ e ci avviciniamo ad un dato più utile.

2.3 Grafo di una settimana di e-mail

Dalla prima settimana registrata dai nostri mail server otteniamo una lista di 37200 account che hanno ricevuto ed inviato almeno una e-mail. Tra questi, ben 6600 ne hanno inviata e ricevuta *una sola*: probabilmente sono esterni alla nostra rete e compaiono in una sola comunicazione. Tenendo conto di questo, rimuoviamo tutti quelli che non hanno inviato/ricevuto almeno 10 e-mail: rimangono nel campione 9578 indirizzi. Pochi utenti sospetti hanno scritto e/o ricevuto migliaia di e-mail.

Ci sono modi più semplici per rappresentare questi dati? Grazie a Mathematica e alle associazioni, siamo facilmente riusciti a costruire un grafo orientato in cui gli indirizzi e-mail compongono i vertici mentre le email scambiate tra questi formano ovviamente gli spigoli (il grafo è orientato, “*A scrive a B*” è diverso da “*B scrive a A*” perciò gli spigoli sono in realtà frecce). Le figure 2.2, 2.3 sono state costruite con 100, 1000 e 7500 e-mail rispettivamente. Dell’ultima (Fig. 2.3) abbiamo tenuto solo il corpo centrale eliminando tutti i gruppi minori, simili a quelli delle prime due immagini anche per dimensioni (persino su 7500 e-mail, i grafi secondari più grandi contano una ventina di vertici, essendo tutti dovuti ad e-mail inviate a una ventina di destinatari).

⁵Si veda la figura 3.2 nel paragrafo “Istogrammi”

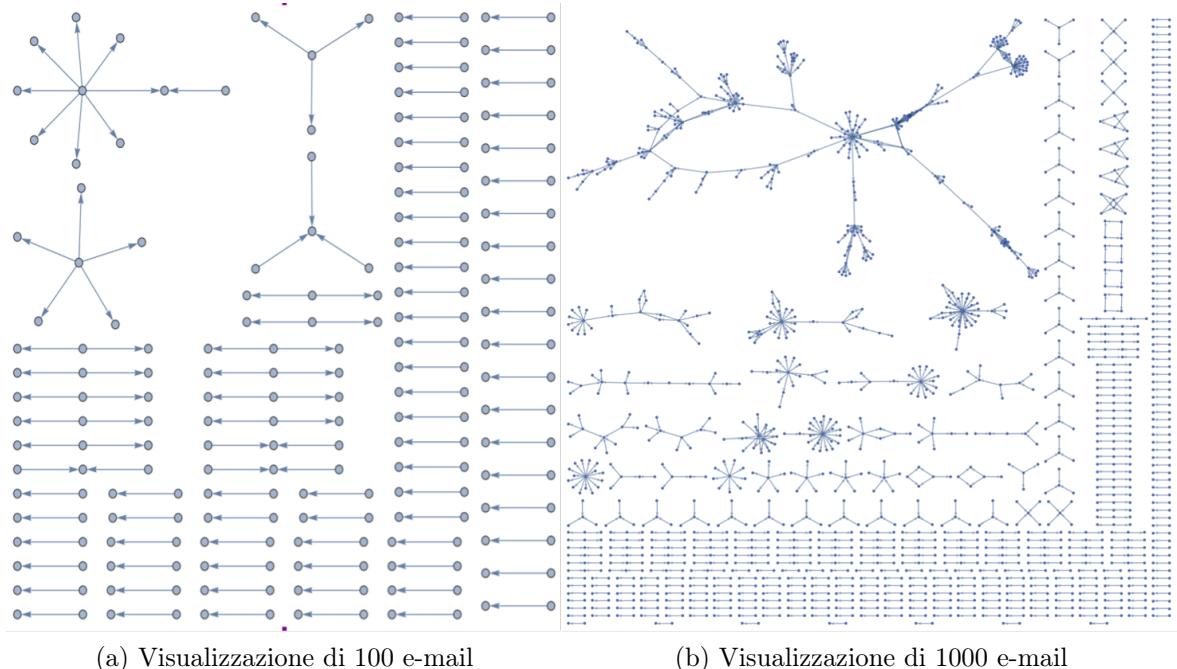


Figura 2.2

Finalmente otteniamo un'immagine simile a quella che avevamo in mente pensando alla nostra rete di utenti. Si distinguono chiaramente utenti “centrali”, connessi alla maggior parte degli altri utenti da uno o pochi spigoli, e altri molto periferici. Addirittura il “braccio” più lungo del grafo, tagliato sulla destra, “dista” dal centro ben undici collegamenti! La posizione di questi utenti estremi è comunque molto instabile per via delle ancora troppo poche comunicazioni visualizzate. Mathematica computa grafi anche con molti più vertici/utenti, tuttavia non ne permette una realizzazione grafica.

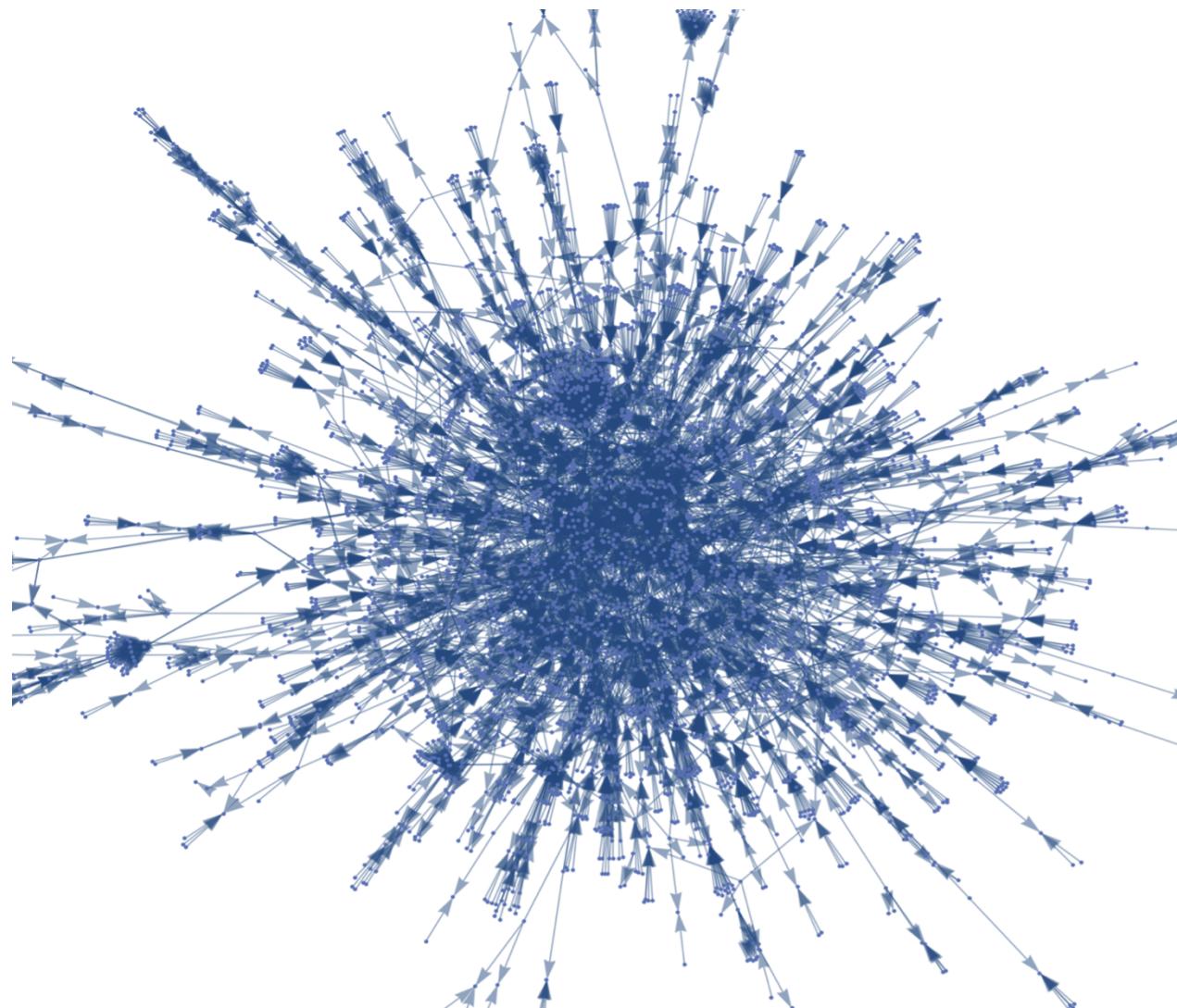


Figura 2.3: Visualizzazione parziale di 7500 e-mail in un grafo

Capitolo 3

Elaborazione dei dati

3.1 Utenti

Abbiamo eliminato i problemi “tecnici” ma non siamo ancora sicuri di aver a che fare con comunicazioni reali o comunque interessanti per il nostro studio. Come si fa a distinguere i messaggi scritti da una persona da quelli generati automaticamente da una macchina? Sappiamo che all’interno di un’università non sarebbe strano vedere email inviate a migliaia di persone contemporaneamente, per esempio i messaggi del rettore o quelli della segreteria. È risaputo, d’altra parte, che in una rete di posta elettronica sono presenti molti utenti “non umani”: server, router, calcolatori che coordinano il traffico di messaggi inviati e ricevuti, inoltrati a due o a cento altri utenti. Considerati i nostri obiettivi, il primo criterio di selezione è che ognuno abbia almeno una email inviata e almeno una ricevuta, cioè che sappia sicuramente leggere e scrivere. Svolgere questo tipo di analisi costa molto tempo in termini computazionali se si parla di milioni di messaggi. Dalla nostra lista di email, passiamo ad una rappresentazione in cui ogni elemento fondamentale è un utente in quanto insieme dei messaggi da questo inviati e ricevuti, eventualmente con i suoi diversi indirizzi/alias. Analizzando anche una sola settimana di comunicazioni si possono trarre molte conclusioni su ciascuno degli utenti che hanno interagito con uno dei tre mail server. Per esempio è possibile individuare un gruppo “piccolo” di utenti con un certo rapporto tra numero di email inviate e numero di ricevute, o tale che la somma di questi ultimi numeri, da noi definita come *attività*, sia abbastanza grande, secondo quelle norme che riteniamo proprie di una persona interna alla nostra rete. Dai dati registrati dai tre MTA *bronze*, *silver* e *gold* durante una settimana campione si ricavano più di un milione di e-mail con una media di 1.4 destinatari effettivi ciascuna. Si possono associare ad ognuno degli

utenti interagenti un indirizzo principale e i suoi eventuali alias, calcolare il numero di e-mail ricevute ed inviate ed elaborare questi dati anche da un computer portatile. Abbiamo misurato il volume di e-mail in entrata e in uscita di ognuno, sia con un contatore semplice che con un peso logaritmico⁶; per esempio vale:

$$(peso \ email \ inviata) = 1 + \text{Log}(\text{numero di destinatari effettivi}) \quad (3.1)$$

Definiamo la somma delle email inviate da ogni utente, pesate secondo la 3.1, come numero efficace di email inviate; similmente, il peso di un messaggio ricevuto è tale da renderlo trascurabile se questo è stato inviato a molti altri utenti; il valore efficace di una sola email è dato da:

$$(peso \ email \ ricevuta) = \frac{1 + \text{Log}(\text{numero di destinatari effettivi})}{\text{numero di destinatari effettivi}} \quad (3.2)$$

Allora il *numero efficace di email ricevute* da un utente sarà la somma di tutti questi contributi durante l'intera settimana. Definiamo inoltre l'*attività* di un utente considerando più importanti le email inviate rispetto a quelle ricevute. In particolare vale:

$$(attività \ utente) = \frac{2}{3}(\text{numero eff. di ricevute}) + \frac{4}{3}(\text{numero eff. di inviate}) \quad (3.3)$$

Consideriamo infine il rapporto tra il numero *assoluto* di email inviate e quello di email ricevute, $r = \frac{(\text{inviate totali})}{(\text{ricevute totali})}$. Si osserva che nel nostro campione molti utenti hanno un rapporto esattamente pari ad uno, mentre altri hanno frazioni semplici ($\frac{2}{3}, \frac{1}{3}, \frac{1}{4}, \dots$). Questo non sorprende: si potrebbe pensare p.e. ad un router che inoltra ogni e-mail che riceve, ma, come detto nella [2], sono in realtà quasi tutti utenti con un'attività molto bassa, corrispondente a piccoli volumi di posta in ingresso e in uscita. Nella selezione dei nostri utenti tipo trascureremo queste persone poco attive e presumibilmente esterne alla nostra rete. Iniziamo subito chiedendo che ognuno dei nostri utenti abbia sia inviato che ricevuto almeno una e-mail, cioè eliminando tutti gli indirizzi che non compaiono mai nel campo *from* o non compaiono mai nel campo *to*. Definiamo infine ρ come rapporto tra i numeri *efficaci* di e-mail inviate e ricevute, dando cioè un peso riscalato logaritmicamente ai messaggi con numerosi destinatari effettivi.

$$\rho = \frac{(\text{inviate efficaci})}{(\text{ricevute efficaci})} \quad (3.4)$$

⁶Il logaritmo è in base dieci. In questo modo una e-mail inviata ad un destinatario ha un peso unitario, una e-mail inviata a due ha un peso di 1.3, una e-mail inviata a dieci ha peso 2, etc.

Per determinare con migliore precisione quali fra gli utenti della settimana scegliere come campione per analizzare tutte le e-mail a nostra disposizione, di ognuno di loro salviamo anche una media del peso in KB di tutte le e-mail inviate e il numero massimo di destinatari raggiunti con una sola comunicazione.

```

RicT = ricA[u]; (*mail ricevute*)
RicP = ricPA[u]; (*mail ricevute pesate con nrcpt*)
Att = K1 * uInvP + K2 * uRicP; (*attività*)
Rho = uInvP / uRicP;
MeanSize = Mean[1.*edb[[All, posSize]]];
MaxRcpt = Max[edb[[All, posNrcpt]]];
Line = {u, uAtt, uInvT, uInvP, uRicT, uRicP, uRho, uMaxRcpt, uMeanSize, uAlias, nAlias};
(*questa è la struttura della riga del nuovo file associata all'utente u*)

```

Figura 3.1: Frammento del programma *Utenti*

Come mostrato in figura 3.1, schediamo tutti gli utenti in una lista di liste chiamata *uLine*, una macrostruttura di cui ogni elemento contiene:

1. Indirizzo;
2. Attività;
3. Numero assoluto di e-mail inviate;
4. Numero efficace di e-mail inviate;
5. Numero assoluto di e-mail ricevute;
6. Numero efficace di e-mail ricevute;
7. Rapporto ρ ;
8. Numero massimo di destinatari raggiunti con una e-mail;
9. Peso medio in B (byte) delle e-mail inviate;
10. Lista di alias;
11. Numero di alias.

3.2 Istogrammi

Per visualizzare queste nuove grandezze, ricaviamo degli istogrammi dai dati a nostra disposizione. Per esempio, verifichiamo che nel nostro campione gli alias siano effettivamente usati e che il nostro programma li riconosca come tali (Fig. 3.2).

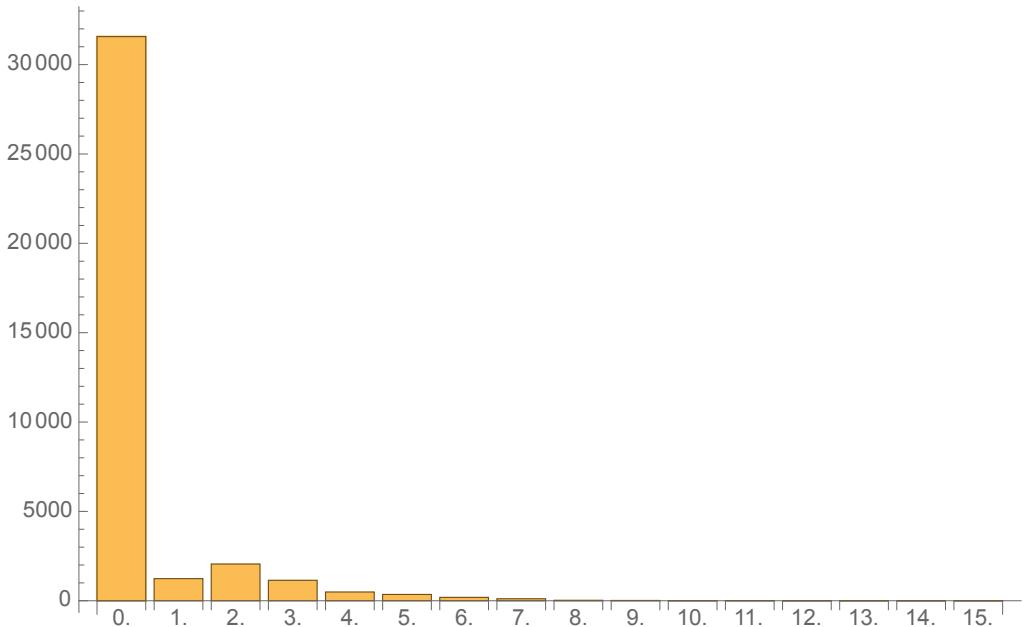


Figura 3.2: Numero di utenti cui sono associati x alias

Se più dell'ottanta per cento dei nostri utenti risulta senza un alias, rimangono in migliaia ad usarne due o tre. L'istogramma dell'*attività* degli utenti non è molto diverso. È evidente che una scala lineare è insufficiente: ci sono tantissimi utenti, quasi tutti, con una attività “bassa” e pochissimi utenti con attività molto alta. Questo andamento ci ricorda quello di una *legge di potenza*: tracciamo il grafico con sia le ascisse che le ordinate riscalate logaritmicamente (in base 10). L'istogramma log-log dell'attività registrata in una settimana è riportato in figura 3.3:

Se la relazione tra il numero di utenti con una certa attività e il valore di questa fosse una legge di potenza, allora il grafico log-log restituirebbe un andamento lineare. Evidentemente non è questo il caso, infatti è ben visibile un andamento costante nella parte centrale. Da questo si vede che all'incirca lo stesso numero di utenti, un migliaio, hanno una attività compresa tra circa 15 e 250. Un risultato simile, cioè un “altopiano” largo un’ordine di grandezza nel centro dell’istogramma, era stato trovato nella [2] considerando

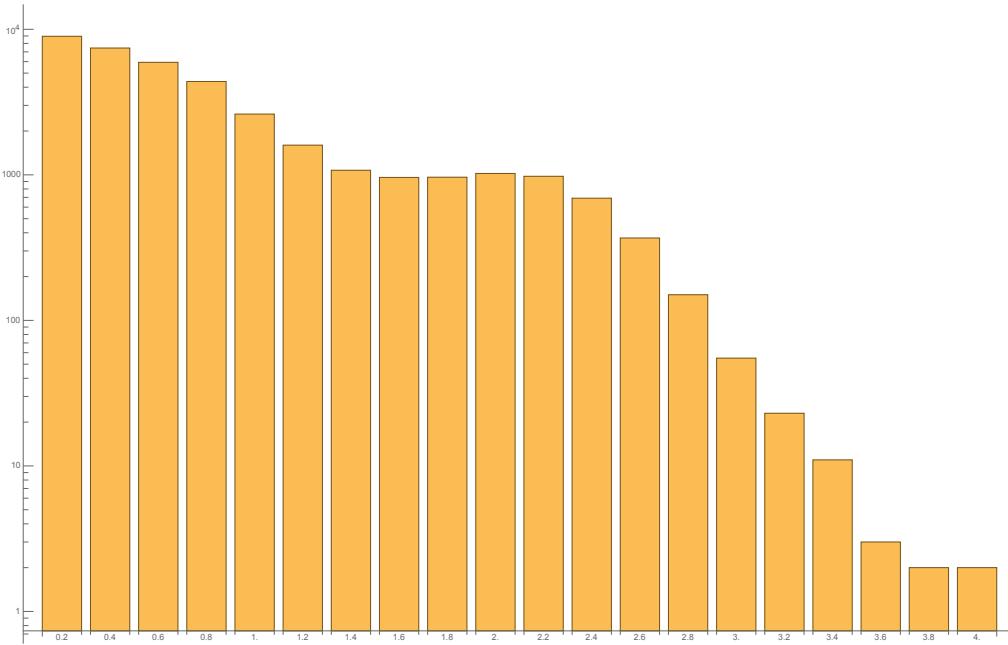


Figura 3.3: Istogramma Log-Log dell'attività

le sole e-mail ricevute⁷. Come ultimo esempio vogliamo mostrare il peso medio in byte delle e-mail inviate, che include tutto il file di testo e gli eventuali allegati. Riportiamo anche qui (in figura 3.4) un istogramma log-log perché altrimenti pochi utenti con una *meansize* molto alta renderebbero il grafico quasi un'unica colonna. In questo caso il minimo è di circa 350 B per e-mail, mentre la media massima supera i 10^7 B cioè 10 MB. È difficile commentare una forma di questo tipo, vagamente simile ad una campana. Si vede però che quindicimila utenti, cioè quasi la metà del nostro campione, inviano mediamente e-mail con un peso compreso tra i 4KB e gli 8 KB. Poco più di millecinquecento utenti inviano e-mail più leggere e un gruppo ugualmente numeroso, visibile nelle ultime cinque colonne del grafico 3.4, invia mediamente files da oltre 2MB.

⁷Secondo la [2], questo sarebbe dovuto agli utenti con un'attività spropositamente alta e che scrivono a "troppe persone", p.e "figure istituzionali". Ciò si potrebbe facilmente verificare filtrando gli utenti con un passa-basso che ne limiti le e-mail inviate nella settimana e svolgendo un goodness-of-fit test che confronti la distanza tra il grafico Log-Log che si ottiene ed una retta opportuna

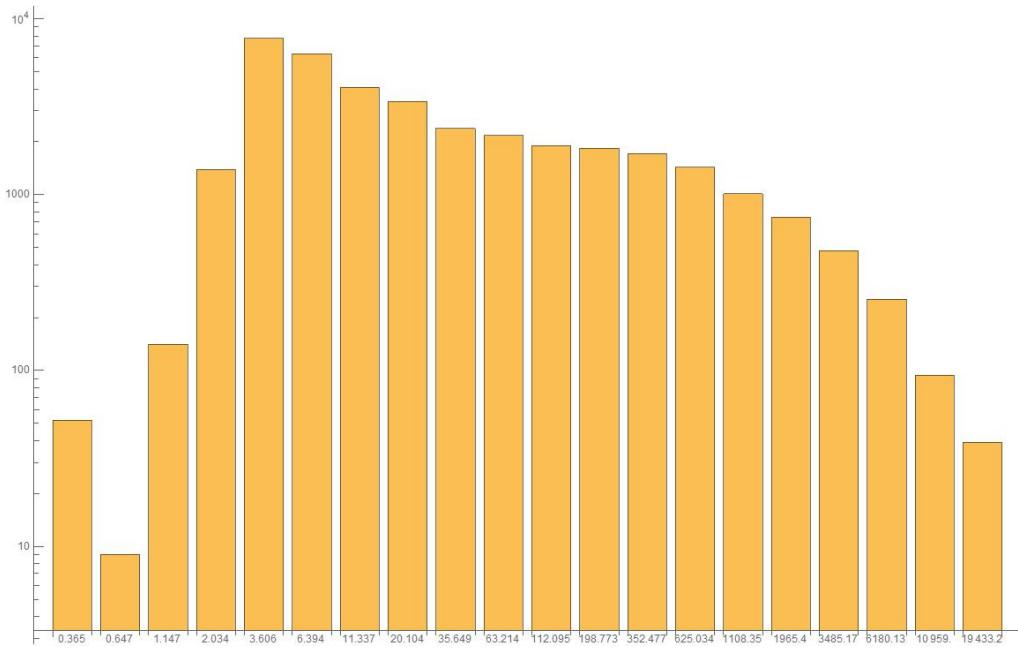


Figura 3.4: Istogramma Log-Log del peso medio in KB x delle e-mail inviate da un utente

3.3 Selezione utenti

Dato il nostro campione di utenti attivi durante una generica settimana, vogliamo ora applicare una selezione per scegliere di *chi* studiare le e-mail nell'arco di sei mesi. Per pescare una decina di indirizzi su trentasettemila, la matematica e Mathematica® sono ottimi strumenti ma non può mancare del “colore” dato da noi nel giudicare quali utenti siano i più giusti da scegliere. Tra i parametri analizzati vi sono l’attività, il rapporto ρ inviate/ricevute⁸, la dimensione media della email e il numero di alias. Per “dare un voto” all’intero campione sulla base di un criterio semplice, è sufficiente un tempo di elaborazione dell’ordine di un secondo. Chiediamo, per iniziare, che ognuno abbia un rapporto ρ compreso tra 0.1 e 1, cioè che non invii più e-mail di quelle che riceve. In questo modo gli utenti risultano meno di diecimila e l’attività media settimanale cresce da 34.0 a 46.9. Se imponiamo anche che il numero massimo di destinatari raggiunti con una sola e-mail stia sotto la soglia dei 50 e che il peso medio delle e-mail inviate superi i 4KB, ci rimangono quasi 8900 utenti. A questo punto vogliamo dare a tutti loro un voto: costruiamo una gaussiana valutata sulla loro attività, intorno ad una

⁸Ricordiamo che calcoliamo ρ riscalando logaritmicamente il peso di una e-mail se questa è inviata a molti destinatari, come mostrato alla fine del capitolo 2.

media di 300 e con una deviazione standard di 100. Questa distribuzione è utilizzata come peso per un'estrazione pseudo-casuale dal nostro campione. Troviamo un insieme di $n = 10$ utenti: la lista è rappresentata in figura 3.5, in cui sono mostrati gli indirizzi, le attività e i rapporti ρ .

```
{486bd501915311d5eff81725f8653279, 921be58f5f6bd7f85ec47754d5c0d0f3,
22c510c8c5213f19224c1be240d9867f, 40c985ec36ea4bf1b8b772f982e4bf7a,
9960fe9a4615de8c3be5588c3ed6d826, 549bfcdaac005f3f66aba5882d60421,
ba107f0cc113319129c0603d2d9e1a2b, 5be46b4adb44df3125ba74761b5b30dc,
99928790b61f36dd6ea9778efe109e1a, dad827fb1e8eaf503bf64c94767e5435}

{277.408, 270.196, 347.478, 211.307,
327.628, 130.115, 123.18, 337.26, 272.436, 169.101}

{0.390958, 0.301963, 0.109053, 0.180837,
0.639699, 0.232782, 0.145034, 0.175174, 0.128008, 0.199554}
```

Figura 3.5: Utenti selezionati, attività e rapporti ρ corrispondenti (una settimana)

Selezionati questi indirizzi -e tutti gli alias che corrispondono loro- possiamo finalmente fare un'analisi più rapida su tutto il dato a nostra disposizione. Considereremo tutte le e-mail in cui tali utenti compaiono come mittente o destinatario nell'arco di sei mesi.

3.4 Codice per sei mesi - *secondoDatiOrdinati*

Il nostro database completo è composto da 150 file di testo dal peso medio di 250MB. Si tratta di oltre cinquanta milioni di e-mail da valutare, tra tutte le registrazioni dei nostri tre mail server *bronze*, *silver* e *gold*. Decidendo di analizzare il solo comportamento di una decina di utenti, è possibile vagliare tutti questi dati in tempi ragionevoli. Grazie a Mathematica possiamo facilmente selezionare solo i dati interessanti: ponendo in serie tutti i file in ordine temporale, salviamo solo le righe in cui almeno uno dei nostri utenti eletti compare come mittente o destinatario con il suo indirizzo principale o con uno degli eventuali alias. Similmente a quanto visto nel capitolo 2 in tutte le operazioni di pulizia, teniamo di questi solo i messaggi relativi all'attività di *demoni* postfix rilevanti (*qmgr*, *local*, *smtp*, ...). Chiamiamo le sopravvissute, per comodità, e-mail "avvincenti". Dopo averle raccolte in un file di testo, possiamo dare quest'ultimo in pasto ai programmi già visti in modo che le e-mail vengano riordinate in strutture opportune. In sei mesi di registrazioni

dei tre mail server, le e-mail che interessano i nostri dieci eletti coinvolgono un totale di 2532 utenti che hanno ricevuto e inviato almeno una e-mail. Verifichiamo se e come sia cambiato il comportamento dei nostri utenti eletti rispetto alla prima settimana di attività riferendoci alla figura 3.5: è ovvio che l'attività misurata su sei mesi dev'essere all'incirca ventiquattro volte quella misurata in una settimana, e così torna. Anche i rapporti ρ inviate/ricevute, la cui misura su una settimana è piuttosto instabile, restano in un range ragionevole con errori che non superano il 30%.

I messaggi inviati in questione sono in tutto 773347. Questo numero molto grande è dovuto quasi per intero a pochi utenti che hanno inviato molti messaggi a migliaia di destinatari⁹, come mostrato dalla rappresentazione in scala Log-Log dell'istogramma relativo al numero di utenti con un certo numero di e-mail "avvincenti" inviate in totale (figura 3.6). In molti inviano poche e-mail e in pochi ne inviano a esagerazione: comportamento tipico di una legge di potenza, tanto è vero che il nostro grafico Log-Log è molto simile ad una retta decrescente.

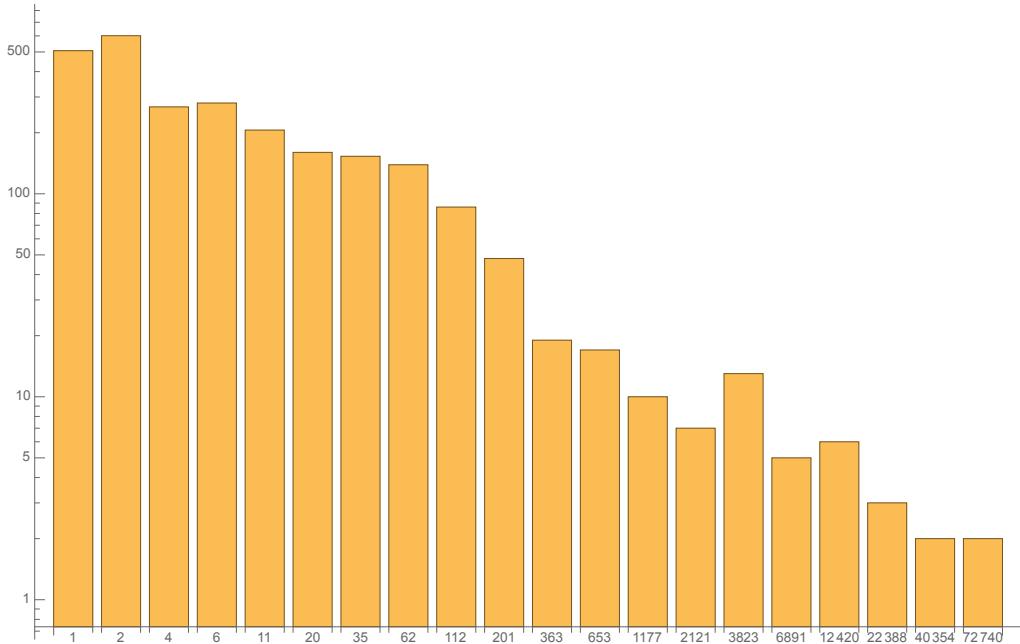


Figura 3.6: Istogramma Log-Log sul numero x di e-mail *avvincenti* inviate in sei mesi da y utenti

⁹26 degli utenti schedati inviano e-mail con un numero di destinatari effettivi (per cui lo status della e-mail, che ha superato i nostri controlli, è "sent") compreso tra 3000 e 6000. Se sfruttiamo la definizione data nell'equazione 3.1, dando poca importanza alle e-mail di tipo *spam*, troviamo che queste oltre 770000 e-mail inviate pesano come se fossero 43300.

Capitolo 4

Ricerca di leggi

4.1 Il tempo

Come già discusso nelle tesi [2],[3], disegnando il numero di email inviate nel tempo si riesce a risalire al giorno della settimana e all'ora, o anche, su tempi più lunghi, individuare periodi festivi. Presa la prima settimana come campione, l'istogramma delle email inviate dovrebbe essere sufficiente per capire in che giorno nel nostro log è stata registrata la prima e-mail. Dal grafico in figura 4.1 è evidente che si tratta di una domenica.

Poiché dal nostro file log troviamo che questa prima email è datata *Oct 2 06:30:34*, troviamo conferma del fatto che risalga al 2016, quando il due di ottobre cadeva di domenica. L'alternarsi dei picchi scandisce i giorni e le notti. Con uno zoom sulla giornata di lunedì e a partire dalle 6:30, si ottiene l'istogramma in figura 4.2.

Il massimo scambio di email è tra le 10:30 e le 12:30, seguito da un minimo locale di due ore in corrispondenza della pausa pranzo. L'attività diminuisce durante il pomeriggio e raggiunge un minimo locale tra le 20:30 e le 22:30, per poi diminuire ancora dopo una leggera ripresa. Nel dominio del grafico, le “25” rappresentano l’una del mattino del giorno successivo e così fino alle “30” (le sei di martedì 3 ottobre). Con questi strumenti abbiamo cercato di capire qualcosa in più sulle e-mail inviate ad un grande numero di utenti; si può realizzare un grafico tridimensionale in cui $z(x, y)$ è il numero di email sul dominio (x, y) dato rispettivamente dal numero di ore passate dall'inizio settimana (domenica, 6.30) e dal numero di destinatari effettivi. Visto dall'alto, risulta come in figura 4.3: non si hanno momenti degni di nota neanche visualizzando i dati di più settimane; tendenzialmente il lunedì rimane il giorno preferito dei messaggi che raggiungono più di cento destinatari.

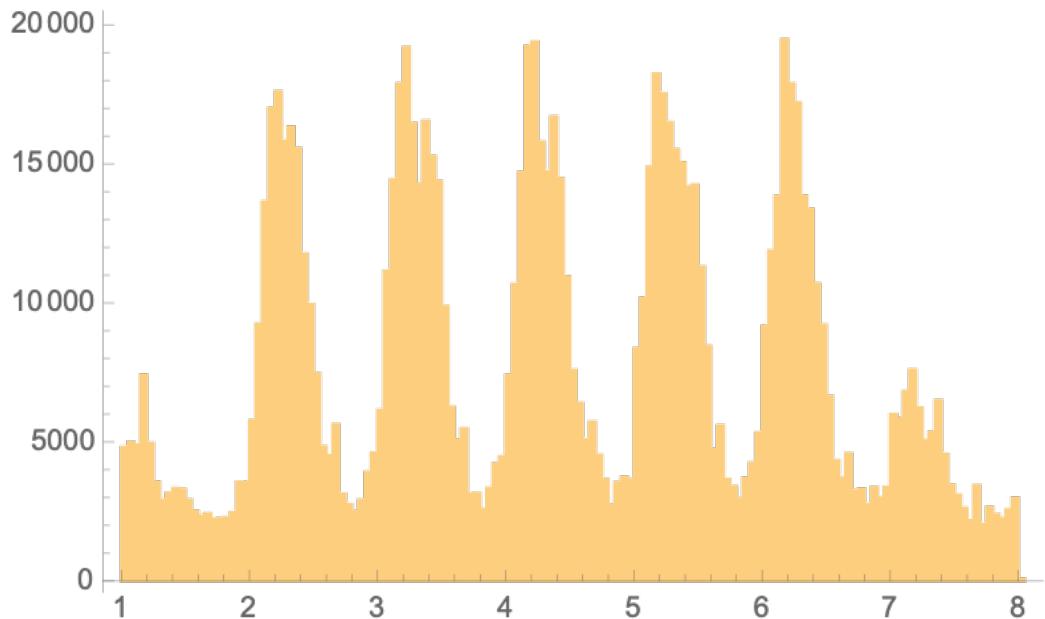


Figura 4.1: E-mail inviate durante una settimana

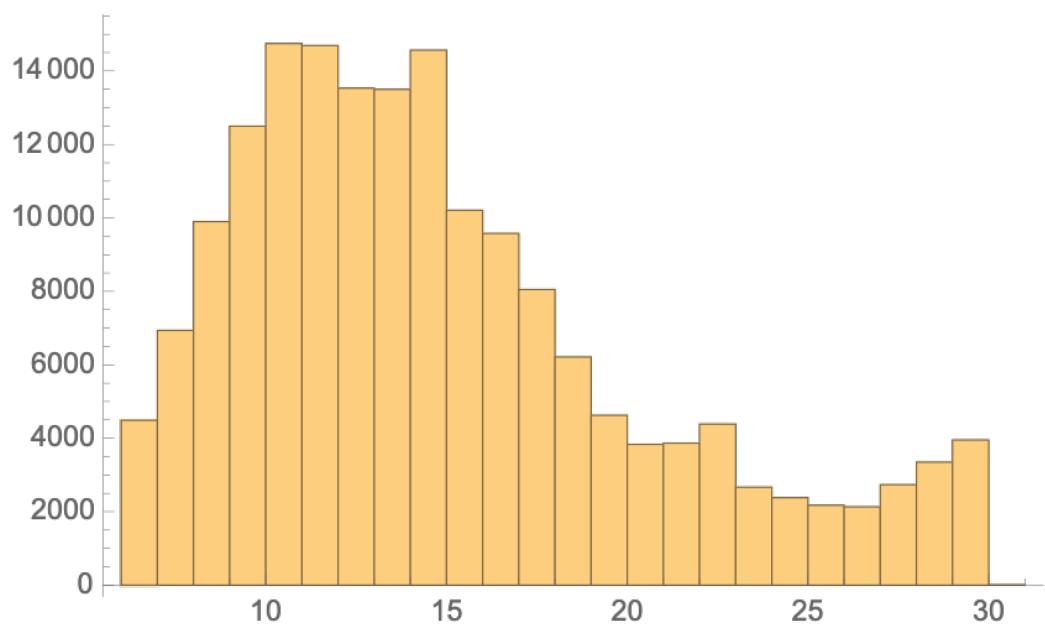


Figura 4.2: E-mail inviate durante una giornata (lunedì)

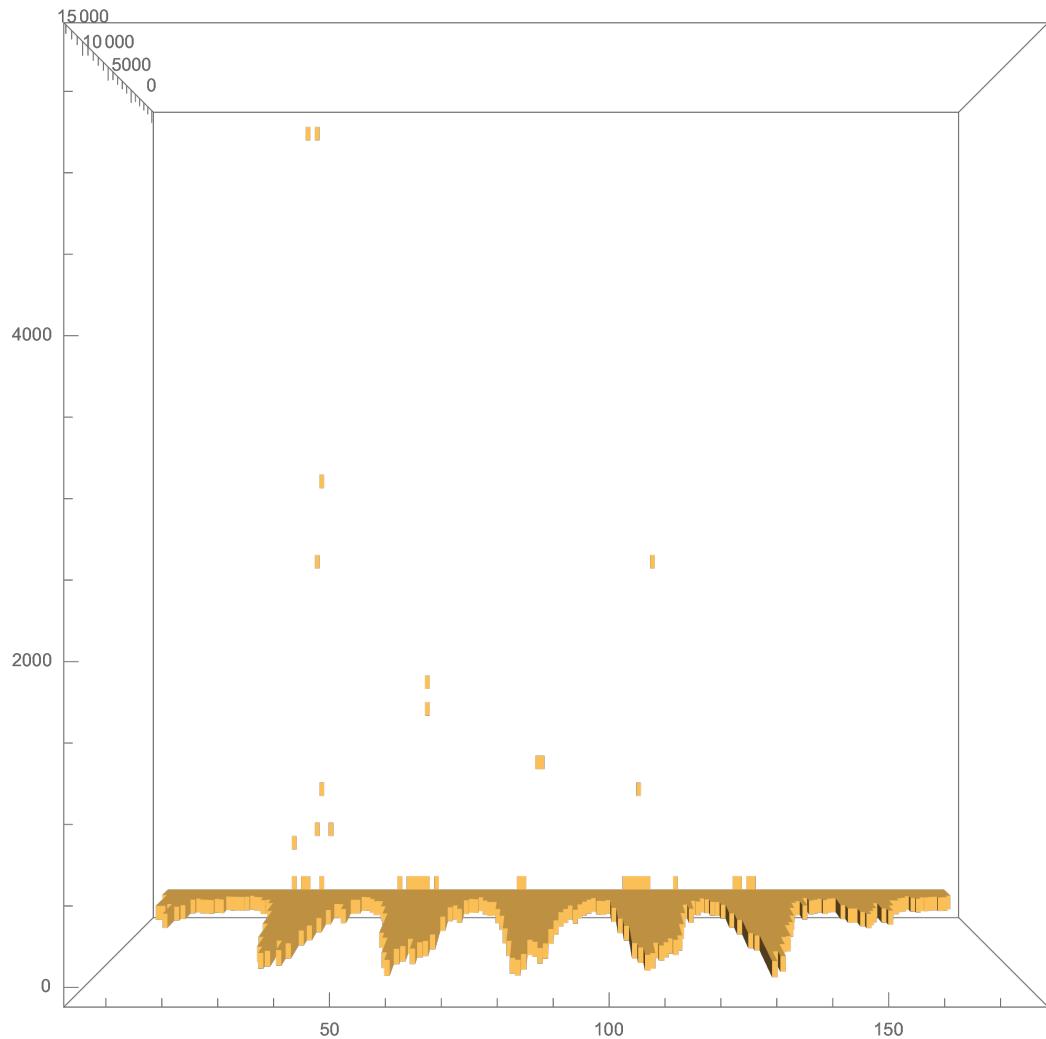


Figura 4.3: E-mail inviate nel tempo e in funzione del numero di destinatari effettivi (in verticale lungo la pagina)

Come già accennato nell'introduzione, il tempo è un parametro molto importante per valutare il comportamento di un utente al di là di quello sopra discusso, periodico e proprio di tutto il campione. I nostri dati sono stati generati dai tre mail server bronze, silver e gold nel periodo compreso l'ottobre 2016 e il settembre 2017, estremi inclusi. Ogni messaggio registrato nel file log contiene la data e l'ora con una sensibilità di un secondo. Nella prima analisi di questi, per avere una rappresentazione più comoda da trattare, abbiamo convertito la data in un numero intero che misuri i secondi trascorsi dalla mezzanotte del primo gennaio 2016. Per esempio, *Oct 9 06:29:41* viene trasformato in *24388181*. Per quanto questa conversione renda le date per niente intuitive, in questo modo un calcolatore può memorizzare e riordinare più agilmente milioni di messaggi; inoltre viene semplice calcolare, per esempio, il tempo medio che un dato utente impiega per rispondere ad una e-mail ricevuta, scrivendo un programma di una riga di testo. Considerare solo il tempo reale, nonostante questa conversione in secondi, è comunque poco per caratterizzare i nostri dati. Vediamo per esempio un grafico (Fig. 4.4) che rappresenta la distribuzione di probabilità $P(\tau)$ associata agli intertempi tra due e-mail consecutive inviate da uno stesso utente.

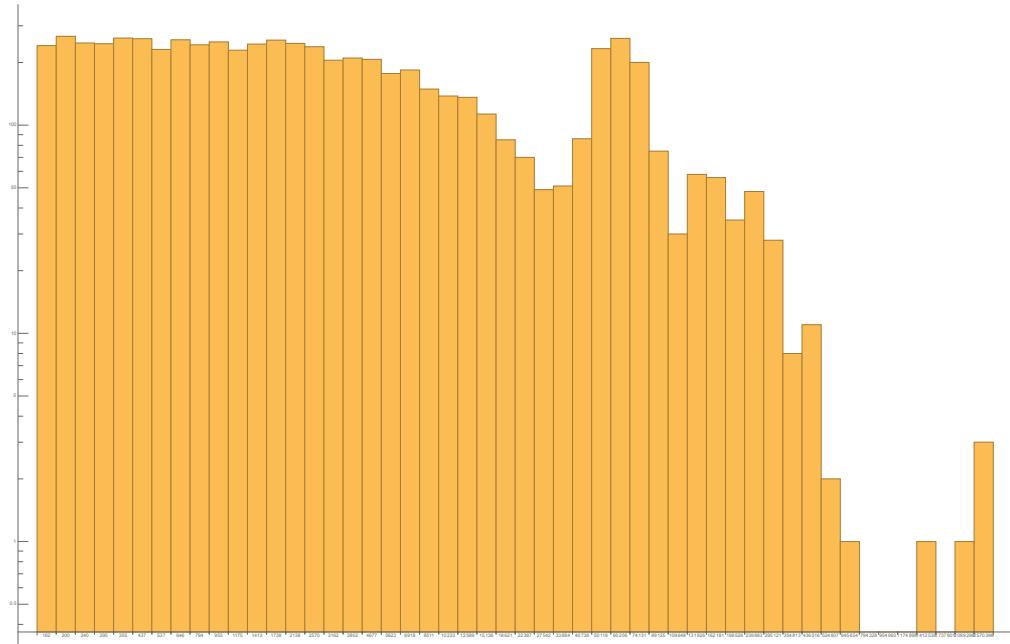


Figura 4.4

Se τ è il valore in secondi che misura l'intertempo tra due azioni di un utente, questo istogramma Log-Log mostra le occorrenze di pause più o meno lunghe $P(\tau)$, da pochi secondi fino a decine di ore, di tutti gli utenti.

La tendenza decrescente mostra che le pause di secondi sono molto comuni, mentre intervalli più lunghi sono sempre più improbabili. Questo andamento è spezzato da un primo picco in corrispondenza di valori per τ nell'intervallo compreso all'incirca tra i ventimila e i settantamila secondi. Come discusso per un grafico analogo nella [3], dividendo tali intertempi per il numero di secondi in un'ora, si trova che essi misurano dalle cinque alle ventiquattro ore: sono quindi pause di una notte o di un fine settimana, “fisiologiche” e periodicamente ripetute. Si capisce così che le statistiche fondate sul tempo reale hanno bisogno di diversi accorgimenti e difficilmente caratterizzano una e-mail. Per riuscire a valutare l'importanza di una risposta non ha senso valutare la giornata dell'utente in tempo reale, troppo complessa per i nostri obiettivi¹⁰. È necessario operare un *re-clocking* fondato sul *tempo proprio* dell'utente, scandito non dai secondi ma dalla sua attività. Con una semplice regola, intuitiva e facilmente implementabile nei comuni linguaggi di programmazione, si ha una nuova prospettiva sui nostri dati, ottima per fare analisi sulle relazioni interpersonali. Considerando che *ogni volta che un utente invia un messaggio il suo tempo proprio cresce di un'unità*, ad ogni utente associamo un orologio diverso da quello di tutti gli altri. Se perdiamo informazioni sulle pause “spontanee”, riusciamo a mettere in evidenza l'attività di scrittura in quanto processo *decision-based*, regolato dalle sole priorità che ognuno di noi assegna ai messaggi di posta e quindi ai nostri interlocutori. Sotto questa luce, parliamo di *tempo proprio*. L'articolo [1] mostra come grazie a questo re-clocking si riescano ad analizzare, con uguali risultati, i diversi tipi di comunicazione scritta (oltre alle e-mail, le lettere e gli sms). In quest'ultimo capitolo useremo il tempo proprio per capire se è vero che le risposte più veloci arrivano a corrispondenti “stretti” cioè più spesso contattati, ed eventualmente se questo comportamento è *universale* o se persone diverse agiscono in modo diverso.

4.2 Gestione delle risposte

Avendo a disposizione tutte le informazioni sulla posta elettronica degli utenti, riordinate come nel capitolo 2 (cioè in una lista con un *idmail*, un mittente e uno o più destinatari, una data, etc.), grazie a Mathematica (si veda la figura 4.5) riusciamo a catalogare ogni e-mail in base alla sua natura di *primo messaggio*, *risposta* o *no-reply*.

¹⁰Si pensi per esempio al caso in cui rispondiamo ad un messaggio importante dopo molto tempo perché, per qualunque motivo, non è stato possibile accedere alla posta elettronica.

```
(*Gestione Risposte di u*)
convUdb = {};
k = 1;
While[k < Length[udb],
  p = Cases[{udb[[k, 2]], udb[[k, 3]]}, Except[u]]; (*partner di u*)
  newUpdb = Cases[udb, {_, u, ToString[p[[1]]], _, _, _, _} | {_, ToString[p[[1]]], u, _, _, _, _}];
  (*updb è la lista di mail scambiate tra u e p*)
  udb = DeleteCases[udb, {_, u, ToString[p[[1]]], _, _, _, _} | {_, ToString[p[[1]]], u, _, _, _, _}];
  newUpdb[[1, 5]] = "Prima Mail"; (*Prima Mail della conversazione tra p e u*)
  For[j = 2, j < Length[newUpdb], j++,
    If[newUpdb[[j, 2]] == newUpdb[[j - 1, 2]], newUpdb[[j, 5]] = "No Reply",
      newUpdb[[j, 5]] = "Answer";
      (*Assegno Answer a email mandate da p che seguono una di u o viceversa*)
      newUpdb[[j, 7]] = newUpdb[[j, 6]] - newUpdb[[j - 1, 6]];
      (*Per una risposta di p il tempo è riferito al tProprio di u*)
    ];
    convUdb = Append[convUdb, newUpdb];
    k++;
  ];
]
```

Figura 4.5: Frammento di codice per la gestione delle risposte e del tempo proprio

Per ognuno dei nostri utenti scelti abbiamo un insieme di conversazioni: l’utente u generico è legato ad una struttura in cui ogni elemento è una scheda associata ad un suo corrispondente p . In ognuna di queste schede vengono registrate tutte le email scambiate tra u e p : tra queste ci sarà una prima e-mail, poi potrebbero esserci delle risposte ma si dà anche il caso in cui le conversazioni sono a senso unico. Ad ogni e-mail associamo quindi una “natura”. Insieme a questa, ne salviamo il tempo (sia proprio dell’utente u che reale) ed eventualmente calcoliamo il “response time”, intervallo di tempo trascorso tra una comunicazione e la risposta a questa, ancora in entrambe le parametrizzazioni. Riordiniamo queste informazioni, come mostrato dal codice Mathematica in figura 4.6, in modo che siano facilmente accessibili.

Tutto questo sarà fondamentale per cercare delle leggi che descrivano l’interazione scritta tra ognuno dei nostri utenti e tutti i loro corrispondenti: considerando insieme il volume di e-mail scambiate e i tempi di risposta, in quest’ultimo capitolo potremo verificare sul nostro dato la validità delle tesi riportate nelle [2],[3].

4.3 Nuvole

Date queste nuove caratterizzazioni di una e-mail, possiamo cercare volti nuovi fra i nostri utenti. Se il *volume in* e il *volume out* sono rispettivamente

```

(* SCHEDE UTENTE:
1 indirizzo utente di u
2 lista di (schede)partner p contattati da u
    SCHEDE PARTNER:
    1 rank (contatore crescente, dal più contattato)
    2 indirizzo di p
    3 volume scambiato con u
    4 schede email
        SCHEDE EMAIL
        1 IDmail
        2 tempo reale
        3 INV/RIC
        4 natura (prima e-mail, answer o no-reply)
        5 tempo di risposta(proprio)
    *)
schedeUtente = {};
For[u = 1, u ≤ Length[conversazioni], u++,
  indirizzoU = indirizziUtente[[u]];
  unaSchedaUtente = {indirizzoU, {}};

  schedePartner = {};
  For[p = 1, p ≤ Length[conversazioni[[u]]], p++,
    indirizzoP = Cases[{conversazioni[[u, p, 1, 2]], conversazioni[[u, p, 1, 3]]}, Except[indirizzoU]][[1]];
    volumeP = Length[conversazioni[[u, p]]];

    emails = {};
    For[n = 1, n ≤ Length[conversazioni[[u, p]]], n++,
      email = Table[0, 5];
      email[[{1, 2, 4, 5}]] = conversazioni[[u, p, n, {1, 4, 5, 7}]];
      email[[3]] = If[conversazioni[[u, p, n, 2]] == indirizzoU, "Inv", "Ric"];
      emails = Join[emails, {email}]
    ];
  ];
]

```

Figura 4.6: Ordinamento di tutte le conversazioni dei nostri utenti

l'insieme delle e-mail ricevute e l'insieme di quelle inviate, possiamo definire il *volume reply* di u come il sottoinsieme del suo volume out che contiene tutte le e-mail di tipo *Answer*, cioè risposte ad e-mail precedentemente ricevute. Si noti che così come non tutte le e-mail inviate sono risposte, l'utente u non dialoga con tutti i suoi corrispondenti: basti pensare ai messaggi di spam o agli indirizzi *noreply*. Considerando solo i corrispondenti p con cui c'è effettivamente uno scambio di e-mail, si può calcolare per ognuno di questi il tempo medio di risposta. Ogni utente p di questo insieme ha cioè il suo caratteristico response time definito relativamente al tempo proprio di u . Come discusso dettagliatamente da Piccini nella [3], realizzando grafici dei volume reply delle conversazioni tra u e p in funzione del tempo di risposta caratteristico di p , si trova un pattern ricorrente in quasi tutti gli utenti u . Riportiamo un esempio in figura 4.7

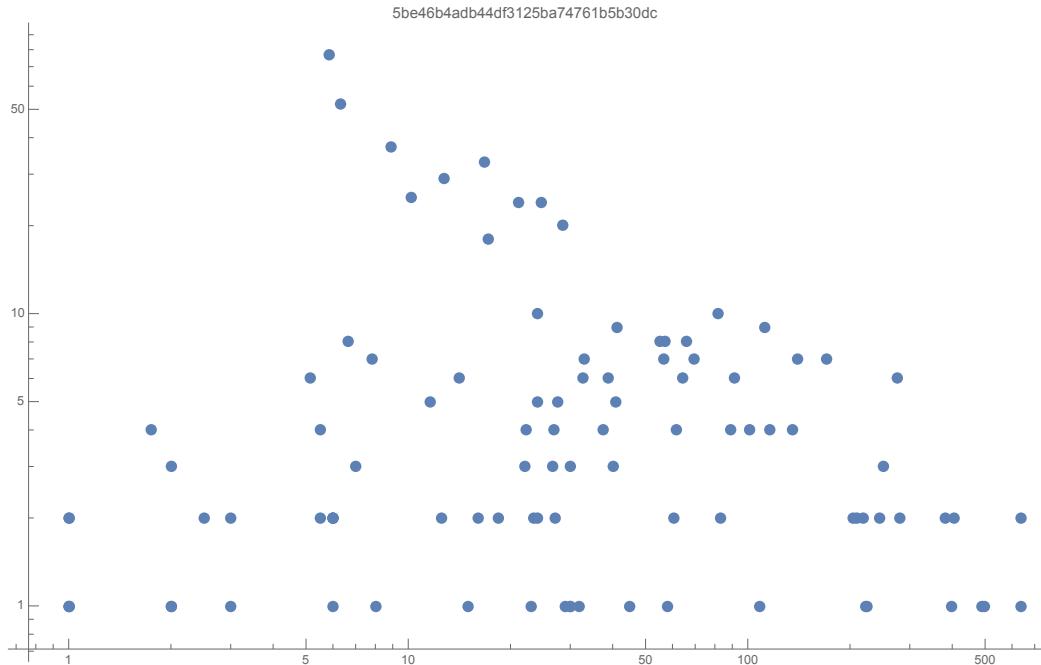


Figura 4.7: Esempio di nuvola

Chiamiamo questi grafici “nuvole” per la loro forma particolare in scala logaritmica: si nota quasi sempre un rettangolo nella parte bassa (volumi piccoli) sovrastato da un triangolo (volumi alti). Pensiamo che la forma triangolare sia legata ad un *andamento decrescente del volume di e-mail scambiato all'aumentare del tempo medio di risposta* in quanto, presumibilmente, l'utente risponde più in fretta ai suoi corrispondenti più “stretti”; inoltre, per costruzione, in un intervallo di tempo limitato non si possono avere corri-

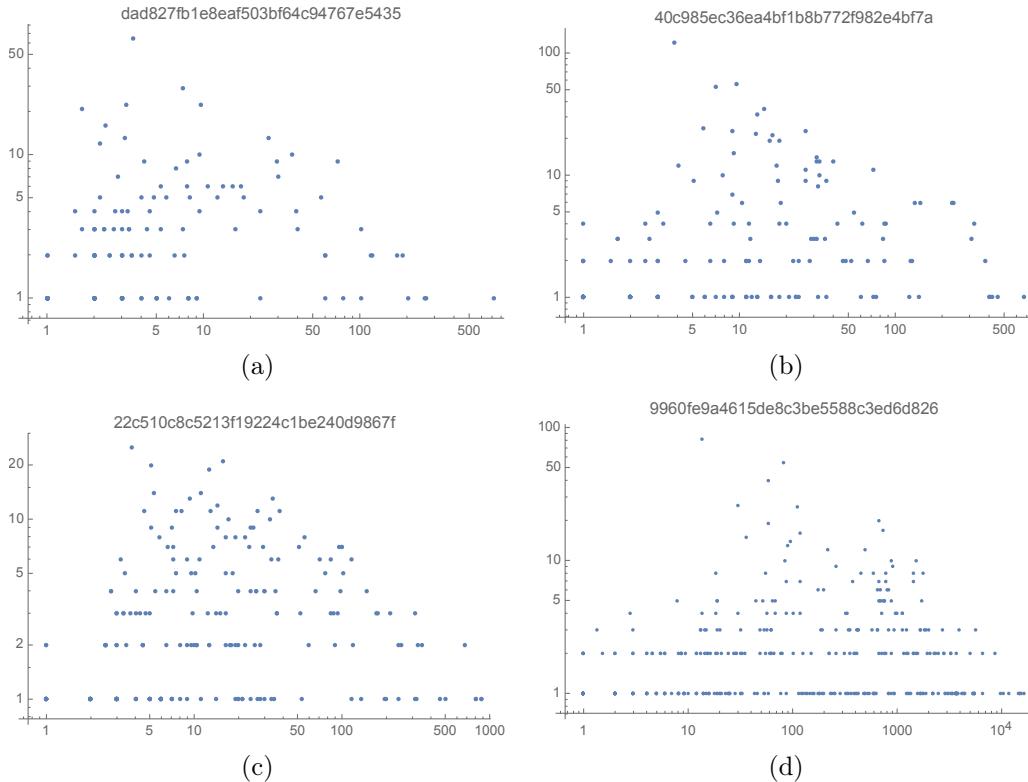


Figura 4.8

spondenti con tempi di risposta molto lunghi e con cui lo scambio di e-mail è fitto. L'apice, che rappresenta l'utente p con il maggior volume reply, non è mai uno degli utenti con il response time minore ma è sempre spostato verso destra. Invece, i corrispondenti con volumi piccoli si distribuiscono uniformemente in tutto l'intervallo di tempi caratteristici costituendo il fondo rettangolare. Questo è coerente con il fatto che comunque, di solito non è al “migliore amico” che rispondiamo con massima urgenza¹¹. Riportiamo in figura 4.8 altre nuvole calcolate sulle e-mail di diversi utenti u.

È evidente la somiglianza tra i grafici rispetto a quanto detto. Le nostre nuvole sono anche simili a quelle mostrate nella [3]. Date queste ricorrenze, abbiamo cercato di evidenziare delle correlazioni tempi/volumi. Non considerando i corrispondenti con bassi volumi, che sono uniformemente distribuiti su tutto il range dei tempi, ma solo quelli più contattati, per i quali abbiamo individuato un andamento dei volumi monotono decrescente, abbiamo calco-

¹¹Per citare ancora la [3], la risposta più rapida la vince *il nostro attore preferito: se ricevessimo anche solo una mail nella vita da un personaggio del genere, certamente la nostra risposta sarebbe quasi istantanea.*

lato l'*indice di Spearman*: misura una correlazione tra due variabili descritta da una funzione monotona. Mentre nella [3] si trovano buoni valori per l'indice, compreso tra 0.3 e 0.55 per quasi tutti i 100 utenti più attivi, i nostri risultati sono molto prossimi allo zero, con un massimo di 0.21: abbiamo cioè realizzato dei test d'ipotesi che mediamente restituiscono un *p-value* troppo alto per rifiutare l'ipotesi nulla di indipendenza tra i vettori discreti dei tempi medi e dei volumi di risposta dell'utente u con i suoi n corrispondenti. Vogliamo qui specificare che il database da noi usato è differente da quello studiato da Piccini, così come lo sono stati alcuni dei procedimenti usati fino ad ora. Per esempio abbiamo implementato un controllo degli alias e abbiamo dato un peso minore alle e-mail inviate a molti destinatari. Inoltre il suo studio è rivolto ai 100 utenti più attivi ed ai 15 meno attivi, mentre noi abbiamo operato una scelta pesata da una gaussiana affinché trovassimo dieci utenti che, in una settimana campione, non avessero troppe e-mail inviate e ricevute, come spiegato nel terzo capitolo¹². Dopo questi calcoli, pensiamo che tale correlazione non sia così *universale* fra gli scrittori. All'analisi di questa dedicheremo anche il prossimo paragrafo.

4.4 Ventagli

Per proseguire nella ricerca di una possibile correlazione tra volumi e tempi di risposta, riportiamo un altro studio della [3] rielaborato sui nostri dati. Un altro pattern ricorrente fra tutti questi messaggi di posta elettronica si nasconde nella famiglia di funzioni di ripartizione (CDF) inverse calcolate sulla distribuzione dei tempi (propri) di risposta dell'utente u generico considerando le e-mail ai suoi diversi corrispondenti p . In particolare, una CDF inversa $P(\sigma)$ rappresenta la probabilità che la risposta arrivi dopo un intervallo di tempo proprio (relativo ad u) *maggior*e di σ . Avendo riordinato i contatti di u assegnando loro un *rank* in base alla frequenza con cui gli inviano e ne ricevono messaggi (dal più attivo al meno attivo), possiamo costruire la famiglia ordinata di distribuzioni $\langle \mathcal{F} = P_i(\sigma), \quad i = 1, 2, \dots, n \rangle$ dove n è il numero totale di corrispondenti di u . Considerando questi n ordinati dal “migliore amico” al meno sentito, $P_i(\sigma)$ è costruita, al variare di i , sul totale

¹²Dalla prima settimana registrata dai nostri mail server otteniamo una lista di 37200 utenti che hanno ricevuto ed inviato almeno una email. L'attività di questi spazia da 1.33 fino a 15570. I nostri 100 utenti più attivi inviano mediamente una e-mail ogni cinque minuti, ventiquattro ore al giorno, per tutta la settimana analizzata e non abbiamo potuto considerarli “buoni utenti”.

delle risposte di u all'insieme dei suoi contatti di rank inferiore ad i^{13} . Ci si aspetta ovviamente che ogni CDF inversa decresca monotona da 1 fino ad annullarsi per tempi di risposta "infinitamente" lunghi. Ma cosa si può dedurre dai grafici al variare del valore i , che limita il rank filtrando quindi intere conversazioni con corrispondenti non vicini e cioè con volumi piccoli?

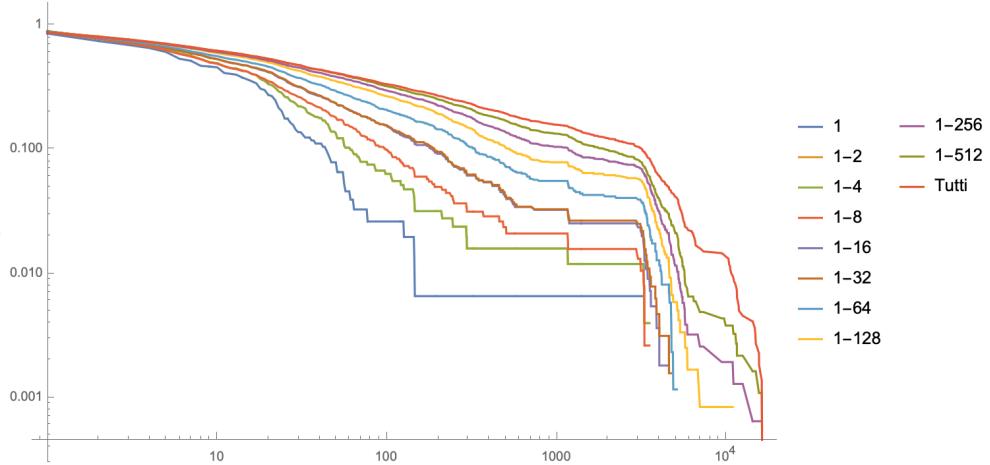


Figura 4.9: Famiglia di distribuzioni cumulate inverse per l'utente `486bd501915311d5eff81725f8653279`

La successione di curve in figura 4.9 è stata costruita per valori di i pari alle prime potenze di due. La più alta, di un rosso acceso, è data invece da tutte le e-mail che interessano u e tutti i suoi n corrispondenti. Al crescere di i , le curve tendenzialmente sovrastano sempre le precedenti, cioè descrivono una probabilità crescente di aspettare tempi di risposta più lunghi. Ciò rispecchia il fatto che, mediamente, questi tempi sono più brevi se il corrispondente è uno con cui si scambiano molti messaggi. Sappiamo che il *migliore amico* non è di solito quello cui sono destinate le risposte più rapide, tuttavia, come visto nelle nuvole, questo record spetta a corrispondenti occasionali che in questo grafico sono rappresentati dalla sola ultima curva, più alta di tutte, insieme a tutti gli altri poco sentiti a cui si risponde in tempi anche molto lunghi. Riportiamo qualche altro esempio in figura 4.10.

Questi disegni, che ricordano dei ventagli, sono ricorrenti nel nostro campione di utenti con poche eccezioni e rafforzano la tesi per cui il tempo di risposta medio decresca con il volume scambiato, come intuito inizialmente dalle nuvole nel paragrafo precedente. Entrambe queste grandezze, del resto,

¹³Cioè la P_1 è misurata sulle e-mail tra il nostro utente u e il suo primo corrispondente nel rank, la P_2 sulle e-mail con i primi due, ..., fino a $P_n(\sigma)$, misurata sul *volume reply* di u verso tutti.

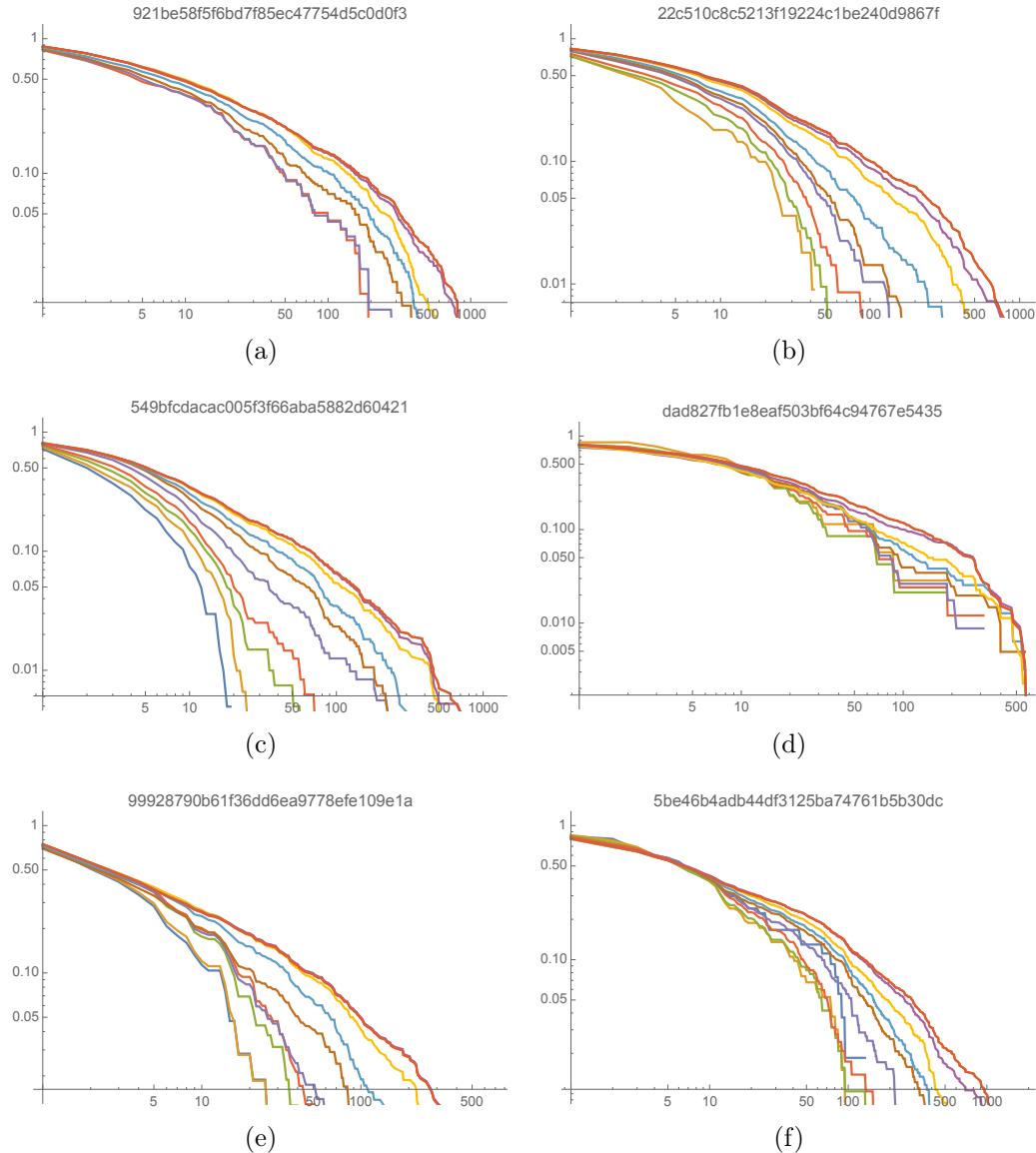


Figura 4.10: Altri esempi di ventagli per diversi utenti del campione

sono caratteristiche di quanto il corrispondente generico ci sia vicino, che sia un'amica o una collega. Concluderemo ora il lavoro rielaborandole nel concetto di *importanza*.

4.5 Importanza percepita, comportamento sociale

La ricchezza del dato rappresentato in forma di *conversazioni* è tale da rendere possibile un'ulteriore analisi, più sociologica e forse utile ad ogni utilizzatore di posta elettronica. Ciò che è presentato in questo paragrafo fuoriesce dall'ambito puramente scientifico, necessitando di ipotesi ad hoc, sostenute solamente da nostre considerazioni, o meglio opinioni, sul comportamento sociale di ogni persona. Siano, a beneficio della fluidità di lettura, Uma e Paloma rispettivamente l'utente u e la partner p. Specifichiamo che, sebbene ogni affermazione relativa ad Uma sia riferibile a qualunque altro utente, i ruoli dei due soggetti non sono interscambiabili: ogni analisi è ripetibile a ruoli invertiti, ma i risultati sono in generale diversi¹⁴. Come è spiegato nei paragrafi precedenti, il tempo che Uma impiega per rispondere a Paloma è legato al tipo di rapporto che intercorre tra le due. Chiamiamo *importanza* la grandezza che descrive la priorità con cui un certo utente risponde ad un altro. In questo senso, la definiamo come l'inverso del tempo (proprio) di risposta. Per ipotesi, Uma percepisce l'importanza che Pamela le mostra, secondo canoni soggettivi che dipendono strettamente da Uma stessa. La tesi dell'analisi è invece: "Uma tende a comportarsi con Pamela nello stesso modo con cui Pamela, dal punto di vista di Uma, si comporta con lei" Traducendo il problema nei termini a cui la tesi si è sempre riferita, specifichiamo che:

- L'importanza che Uma mostra a Pamela si misura a partire dai tempi di risposta delle email inviate da Uma;
- L'importanza che mostra Pamela si misura a partire dai tempi di risposta delle email inviate da Pamela;
- Entrambi questi tempi di risposta sono scanditi dall'attività di Uma, come spiegato all'inizio del capitolo.

Cerchiamo quindi i tempi medi di risposta di Uma verso ogni suo corrispondente e li confrontiamo graficamente (in figura 4.11) con i tempi medi delle

¹⁴Sia i tempi di risposta delle e-mail di Paloma a Uma che quelli da Uma a Paloma sono misurati sulla base dell'attività di Uma, perciò scambiando i ruoli cambia anche la base su cui vengono elaborati gli stessi dati. Se per esempio Paloma inviasse più email al secondo di Uma, allora il tempo proprio di Paloma scorrerebbe più velocemente di quello di Uma

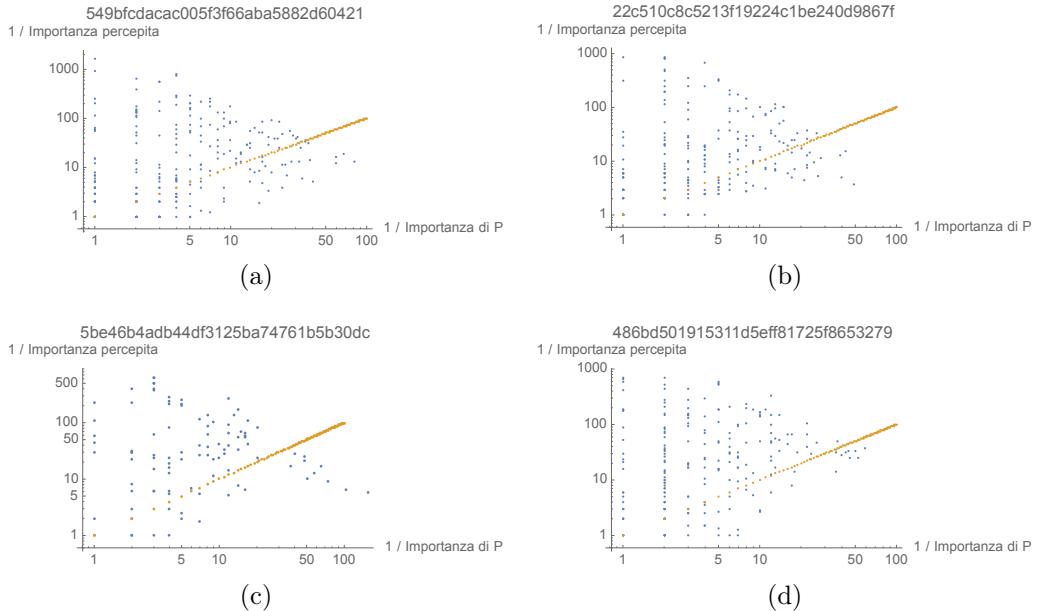


Figura 4.11: Tempi di risposta di p (y) sui tempi di risposta di u (x)

risposte ricevute, come percepiti da Uma. La tesi sarebbe provata se le figure ottenute aderissero alla bisettrice del grafico, riferimento in giallo, ma così non è.

Invece, per tutti i nostri buoni utenti, i grafici mostrano sempre una netta tendenza dei partner a rispondere più lentamente. Nei termini del problema, questo si traduce concludendo che: *chiunque sia Uma, la frustrazione che le genera il relazionarsi con la propria casella di posta è giustificata, poiché in media riceve risposta dai suoi corrispondenti più lentamente di quanto non impieghi a rispondere lei*. In quanto intuitivamente assurda, abbiamo inizialmente rifiutato tali risultati. Ciò che ci stupisce è come un sistema di e-mail chiuso come il nostro possa mostrare un tale sbilanciamento, per ognuno dei buoni utenti valutati. Forse a provocare tale asimmetria è la diversa scala dei tempi che ogni utente fa propria. Ci riserviamo comunque di fare ulteriori affermazioni prima di proseguire nell'indagine.

La nostra tesi iniziale è stata smentita dai grafici 4.11. Vogliamo però verificare che le comunicazioni non tendano alla bisettrice neanche allo scorrere del tempo reale, o che in generale Uma modifichi o meno l'Importanza che dà a Paloma, via via che la relazione tra le due si evolve. Per modellizzare tale evoluzione selezioniamo le coppie Uma-Paloma, con almeno cento e-mail scambiate, così da rendere possibile almeno un tentativo di verifica. Dai nostri dati, otteniamo 8 coppie tipo che soddisfano tale criterio. Le figure 4.12 mostrano sull'asse delle ascisse il tempo reale e sull'asse y il tempo proprio

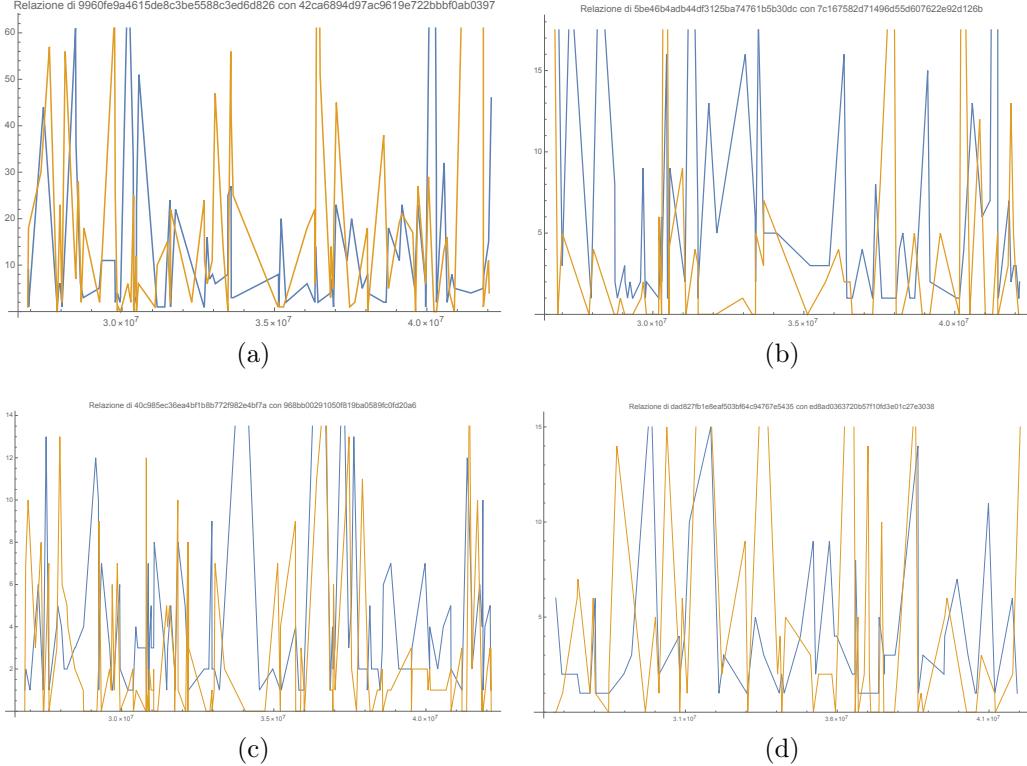


Figura 4.12: Andamento dei tempi di risposta nel tempo reale per diverse coppie utente-partner

di risposta di una e-mail.

I grafici, è evidente, non sono soddisfacenti. L'andamento delle spezzate è fortemente irregolare per ogni coppia Uma-Paloma, i cui tempi di risposta sono rappresentati rispettivamente dai picchi blu e gialli. Notiamo però che i minimi della spezzata gialla (i tempi di risposta di Paloma) sono pari a zero, mentre la spezzata blu (i tempi di risposta di Uma) ha i minimi in uno. Questo off-set, dovuto alla nostra definizione del tempo di risposta¹⁵, farebbe arrabbiare ancora di più Uma per quanto detto a proposito delle figure 4.11: nonostante parta, in un certo senso, già in svantaggio, può sempre aspettarsi tempi di risposta più lunghi di quelli che concede.

¹⁵Per costruzione, alla prima e-mail inviata da Uma corrisponde il tempo proprio $\sigma = 1$, alla seconda $\sigma = 2, \dots$

Chiaramente per Uma non è possibile rispondere in un tempo nullo, in quanto il tempo proprio associato alla sua e-mail è già cresciuto di un'unità. Invece se Uma scrive una e-mail con tempo σ e Paloma le risponde prima che ne scriva altre, il tempo di risposta resta $\sigma - \sigma = 0$.

Capitolo 5

Conclusioni

Riassumiamo qui le conclusioni che già i singoli paragrafi, a cui sono riferite, contengono per esteso. Possiamo affermare che, pur esistendo delle linee guida generali con le quali rappresentare i dati, non abbiamo trovato per vie teoriche metodi universalmente migliori di altri. Un esempio è la definizione di ottimo per il numero di barre in un istogramma. Tale definizione, seppur discussa in letteratura, è rimasta per noi una scelta vaga e adattabile allo scopo contingente della rappresentazione. Due tipi di elaborazioni di dati da cui, per nostra esperienza, non si può certamente prescindere, sono la suddivisione in classi/categorie e il filtraggio. Per entrambe queste rimane primario il ruolo dell'analista, poiché egli dovrà fare delle scelte che condizioneranno i risultati ottenuti. A questo proposito, abbiamo sempre ritenuto importante esplicitare le nostre scelte e le nostre motivazioni. Un esempio importante è stata la necessità di individuare dieci buoni utenti su di un insieme di oltre trentamila. Questa selezione ha forse necessitato il nostro sforzo più grande in termini di opinioni personali.

Sulla presunta correlazione fra tempi di risposta e volumi scambiati abbiamo ottenuto risultati discordanti: i valori ottenuti dai test di Spearman tendono a negare l'ipotesi di relazione decrescente, sostenuta invece dalle evidenze grafiche nelle *nuvole* e nei *ventagli*. Precisiamo che le analisi sulle leggi di potenza a cui la tesi fa alcuni riferimenti e in cui crediamo che, a questo proposito, si nascondano risultati di fondamentale importanza, saranno oggetto di una seconda tesi del laureando Lorenzo Tutolo, con cui abbiamo condiviso gran parte del percorso.

L'ultimo paragrafo, intorno all'*importanza percepita*, è stato un nostro tentativo di andare oltre i riferimenti bibliografici e la letteratura -a noi nota- su questi temi. Il partner *p* generico sembra rispondere molto più lentamente di quanto non faccia l'utente *u* in tutti i nostri grafici. Nonostante la sorpresa, non siamo riusciti ad identificare un eventuale errore nei programmi, né ab-

biamo trovato quella che potrebbe essere una spiegazione valida per questo risultato bizzarro. Sarebbero quindi necessarie ulteriori verifiche dopo aver allargato il campione di utenti e averlo reso più eterogeneo.

Sebbene questa conclusione sia incerta, siamo sicuri che suggerisca un'ottima morale: non è cosa buona stressarsi se ci sembra che la risposta non arrivi. Così, nelle stesse intenzioni con cui era stata introdotta questa tesi, lasciamo la sua conclusione aperta ai colleghi futuri: per un uso più consapevole della posta elettronica.

Bibliografia

- [1] M. Formentin, A. Lovison, A. Maritan, and G. Zanzotto, *Hidden scaling patterns and universality in written communication*, Physical review, E 90, 2014.
- [2] Notarangelo, Barboni, Carella, *Analisi statistica della comunicazione umana scritta*, 2020.
- [3] M. Piccini, *Tratti universali del comportamento umano evidenziati dall'analisi della comunicazione scritta*, 2015
- [4] Wolfram Language & System Documentation Center, reference.wolfram.com
- [5] Postfix Documentation, postfix.org