Universitá degli studi di Padova
Dipartimento di Scienze Statistiche

Corso di Laurea Triennale in
Statistica, Economia e Finanza

Relazione Finale

Topic Model Workout: un approccio per l'analisi di microblogging, mass media e dintorni

Relatore Livio Finos
Correlatore Dario Solari
Dipartimento di Scienze Statistiche

Laureando Ferraccioli Federico
Matricola N. 1033416

Anno Accademico 2013/2014

Indice

Elenco delle figure

Elenco delle tabelle

Caterina Simonsen: La sperimentazione sugli animali mi ha permesso di vivere. Insultata e minacciata di morte

Caterina Simonsen é una studentessa di Veterinaria all'universitá di Bologna. Colpita da quattro malattie genetiche rare, é divenuta il bersaglio di estremisti animalisti su Facebook dopo avere pubblicato una foto che la ritrae con il respiratore sulla bocca e un foglio in mano: "Io, Caterina S., ho 25 anni grazie alla vera ricerca, che include la sperimentazione animale. Senza la ricerca sarei morta a 9 anni. Mi avete regalato un futuro". Dopo le minacce di morte, Caterina ha postato sul social network un video di risposta: "Vi faccio vedere come si vive con le mie malattie, e dopo gli oltre 30 auguri di morte e oltre 500 offese, io metto 'a nudo' la mia realtá perché voi capiate che l'unica mia 'colpa' in tutto ció sia stata 'curarmi' senza uccidere nessuno direttamente", ha commentato.

L'Huffington Post, 28-12-2013

Come tutto ha inizio

La presente tesi ha preso le mosse dal discusso dibattito avvenuto nel periodo che va dall'ultimo trimestre 2013 al primo trimestre 2014, riguardante il caso di Caterina Simonsen. Come si evince dall'articolo appena presentato Caterina Simonsen é diventata bersaglio per la lotta alla sperimentazione animale dopo aver pubblicato una foto su Facebook. Nel web in particolare il caso ha avuto notevole seguito, ed é per questo che si é deciso di raccogliere, nelle prime tre settimane di Gennaio, un dataset di tweet riguardanti l'argomento. Lungi dal prendere una posizione nel dibattito, la tesi sfrutta la possibile polarizzazione di idee e il mix di argomenti per analizzare alcune tecniche di text mining: dalle piú semplici su testi brevi come puó essere un tweet alle piú complesse su interi articoli.

La prima parte presenta l'analisi di un dataset di tweets riguardanti la sperimentazione animale, e cerca di raggruppare e identificare, tramite utilizzo di funzioni di distanza tra testi, i tweets piú rilevanti, con l'obiettivo di sintetizzare il dataset in un insieme ridotto contenente le opinioni piú influenti. La seconda parte prende le mosse dalla prima: vengono estratti i testi degli articoli relativi agli indirizzi url nei tweets precedentemente analizzati e vengono applicati e discussi modelli piú complessi confrontandone i risultati. In particolare: nel primo capitolo verranno introdotte alcune semplici analisi descrittive per comprendere meglio la struttura del dataset e coglierne approssimativamente il contenuto informativo; nel secondo capitolo é presentata una funzione basata sulla distanza tra testi, `RTHound,` per l'individuazione e la clusterizzazione dei retweet; il terzo capitolo é di transizione, i modelli presentati sono piú complessi ma i dati osservati sono sempre i tweet; il quarto ed ultimo capitolo presenta svariati modelli per il text mining, Topic Model, applicandoli non piú a tweet ma ad articoli e documenti. La scelta di percorrere le due strade vuole mostrare la differenza nell'analisi di testi brevi (quali i tweets), nei quali l'informazione é contenuta in un numero limitato di parole, e quella di testi piú corposi (quali gli articoli), in prevalenza caratterizzati dalla presenza di una mistura di argomenti, identificati piú avanti come topic. Se all'apparenza non sembra fondamentale la lunghezza dei testi, nel corso delle analisi é apparso chiaro essere un assunto imprescindibile: da una parte abbiamo i tweet, testi che variano dalle 5 alle 20 parole, caratterizzati da un singolo argomento (e in molti casi addirittura assente, si pensi ad esempio ad un tweet che reindirizzi ad un articolo di un sito esterno) e dalla presenza di simboli provenienti dal mondo del web quali emoticons, hastag etc.; dall'altra testi molto piú articolati, in cui il numero di parole e di argomenti trattati aumenta di molto insieme alla complessitá delle sfumature linguistiche della lingua italiana. Saranno presentate dunque entrambe le possibilitá, per dare una panoramica piú ampia ed esaustiva possibile dei topic model.

Capitolo 1

Processamento e analisi descrittive

In questo primo capitolo vedremo come ottenere un dataset di tweet, quindi come utilizzarlo per trarne informazioni attraverso alcune semplici analisi. Piú in particolare, le distribuzioni giornaliere e orarie dei tweet e dei retweet, gli users piú attivi, i termini piú utilizzati, la caratterizzazione del linguaggio e infine la creazione di una rete di retweet e di parole: la prima collega gli users attraverso i retweet, la seconda riassume le relazioni tra le parole piú frequenti e quindi le co-occorrenze delle stesse. Tutte le analisi di questo capitolo, e piú in generale di tutta la tesi, sono state fatte con il software statistico R; in questo capitolo é predominante l'utilizzo della libreria TextWiller (Solari, Finos, Redaelli, con contributi di Marco Rinaldo, Branca, and Ferraccioli., 2013).

1.1 Estrazione del corpus di tweet

Il dataset utilizzato è composto da più di 3000 tweets in lingua italiana raccolti tra il 2014-01-06 e il 2014-01-28, riguardanti la sperimentazione animale: la scelta è stata fatta attraverso query (Caterina simonsen, #caterinasimonsen, #sperimentazione, #sperimentazioneanimale, 'sperimentazione animale', #iostocongiovanna, #iostoconcaterina, #iostoconlaricerca, #vivisezione, #nazimalisti). Per ottenere il dataset é possibile appoggiarsi alla libreria twitteR (Gentry, 2013), o al dump_tool di Matteo Redaelli (Redaelli, 2014). Il dataset é composto da 21 variabili, ed é ottenibile con i seguenti comandi:

```
data(TWsperimentazioneanimale)
tw=TWsperimentazioneanimale
str(tw)
```

```
'data.frame':        3022 obs. of  22 variables:
 $ text         : chr  "RT @orianoPER: http://t.co/RD5vyvA1Gw
         dr.ssa S. Penco-Ricercatrice-Premio Nazionale 2013 per la
 $ favorited    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ favoriteCount: num  0 12 0 0 0 0 0 0 0 0 ...
 $ replyToSN    : chr  NA NA NA NA ...
 $ created      : POSIXct, format: "2014-01-06 09:16:59"
 $ truncated    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ replyToSID   : chr  "" "" "" "" ...
 $ id           : chr  "420121810793795584" "420212124988616705"
 $ replyToUID   : chr  "" "" "" "" ...
 $ statusSource : chr  "<a href=\"http://twitter.com/tweetbutton\"
 $ screenName   : chr  "momixart" "FedericoSbandi" "alss77"
 $ retweetCount : num  2 8 8 8 8 8 8 8 8 0 ...
 $ isRetweet    : int  1 0 1 1 1 1 1 1 1 0 ...
 $ retweeted    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ longitude    : num  NA NA NA NA NA NA NA NA NA NA ...
 $ latitude     : num  NA NA NA NA NA NA NA NA NA NA ...
 $ ts           : chr  "2014-01-12 10:18:02" "2014-01-12 10:18:02"
 $ lang         : chr  NA NA NA NA ...
 $ sentiment    : int  NA NA NA NA NA NA NA NA NA NA ...
 $ geocode      : chr  NA NA NA NA ...
```

```
 $ lang_twitter : chr  "it" "it" "it" "it" ...
 $ created.day  : POSIXct, format: "2014-01-06" "2014-01-07" ...
```

1.2 Operazioni preliminari

I tweet appena ottenuti possono essere sfruttati per qualche analisi preliminare prettamente descrittiva: vediamo ad esempio le distribuzioni giornaliere e orarie, la diffusione del link più citato, gli hashtag e gli users più citati, la distribuzione nel tempo per le citazioni di un dato utente. Le statistiche descrittive appena accennate sono possibili attraverso la creazione della *Document Term Matrix*, una matrice alle cui righe corrispondono i tweets e le cui colonne sono composte dai conteggi dei termini presenti nei testi (verrá discussa in dettaglio piú avanti).
Partiamo con la pulizia del formato, che ci permette di gestire più agevolmente le date del database estratto da un dump di Twitter:

```
tw=fixTimeStamp(tw)
```

Successivamente la pulizia del testo:

```
tw$testo=normalizzaTesti(tw$text,normalizzacaratteri=TRUE,
       tolower=TRUE,perl=FALSE,fixed=FALSE)
```

L'operazione di normalizzazione del testo permette di condurre le successive analisi più facilmente, eliminando incorrettezze. Nello specifico la funzione permette di normalizzare i caratteri, eliminando quelli non necessari, eliminare la punteggiatura, sostituire gli emoticon e gli indirizzi web con specifiche stringhe, ed eliminare le stopword. Vediamo un esempio, prima e dopo la normalizzazione:

```
"RT @orianoPER: http://t.co/RD5vyvA1Gw dr.ssa
S. Penco-Ricercatrice-Premio Nazionale 2013 per
la #Ricerca- #vivisezione #sperimentazione #animale"

"rt @orianoper wwwurlwww dr ssa s penco
ricercatrice premio nazionale 2013 per la
#ricerca #vivisezione #sperimentazione #animale"
```

La caratterizzazione del linguaggio in Twitter non è sempre precisa, assicuriamoci di estrarre i soli tweet in lingua italiana, avvalendoci della funzione `textcat` presente nella stessa libreria (Hornik, Mair, Rauch, Geiger, Buchta, and Feinerer, 2013). La tabella dei conteggi relativa é la 1.1.

```
textcat(tw$text,ECIMCI_profiles)
```

Lingua	Conteggi	Lingua	Conteggi
Ceco	1	Olandese	3
Danese	1	Norvegese	11
Tedesco	1	Polacco	7
Inglese	47	Portoghese	7
Spagnolo	10	Rumeno	3
Francese	17	Slovacco	4
Croato	5	Sloveno	1
Ungherese	9	Albanese	5
Italiano	2730	Serbo	1
Latino	14	Svedese	13

Tabella 1.1: Conteggi dei tweet per nazionalitá

1.3 Analisi descrittive

Iniziamo ora con qualche analisi descrittiva, nello specifico distribuzioni giornaliere e orarie dei tweet, i grafici relativi sono rappresentati in figura 1.1. Come prima cosa creiamo nuove variabili, che aggiungeremo al nostro dataset, relative al giorno e all'ora di creazione di ogni tweet; queste ci permetteranno di costruire le distribuzioni suddette.

```
tw$created.day=as.POSIXct(round(tw$created,"day"))
tw$created.hours=as.POSIXct(round(tw$created,"hours"))
plot(table(tw$created.hours),ylab="tweets")
summary(as.numeric(table(tw$created.hours)))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   2.000   5.000   7.425   9.000 109.000
```

```
plot(table(tw$created.day),ylab="tweets")
table(tw$created.day)
summary(as.numeric(table(tw$created.day)))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    1.0    66.0   110.0   131.4   142.5   540.0
```

	Conteggi		Conteggi
2014-01-06	1	2014-01-18	113
2014-01-07	47	2014-01-19	65
2014-01-08	137	2014-01-20	67
2014-01-09	219	2014-01-21	64
2014-01-10	166	2014-01-22	87
2014-01-11	127	2014-01-23	131
2014-01-12	70	2014-01-24	110
2014-01-13	280	2014-01-25	82
2014-01-14	540	2014-01-26	19
2014-01-15	320	2014-01-27	85
2014-01-16	148	2014-01-28	14
2014-01-17	130	-	-

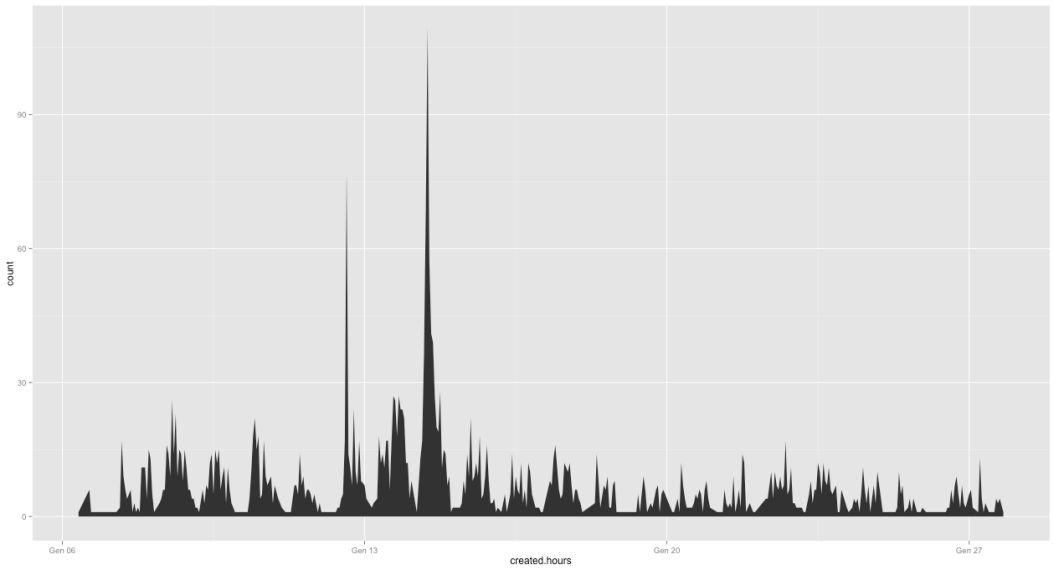
Tabella 1.2: Conteggi giornalieri dei tweet nel periodo considerato

Puó essere interessante vedere anche gli users più attivi e le relative distribuzioni, per carpire informazioni aggiuntive sull'attivitá in Twitter, non limitandoci solo a quelle totali. A questo scopo calcoliamo prima i conteggi degli user (tabella 1.3):

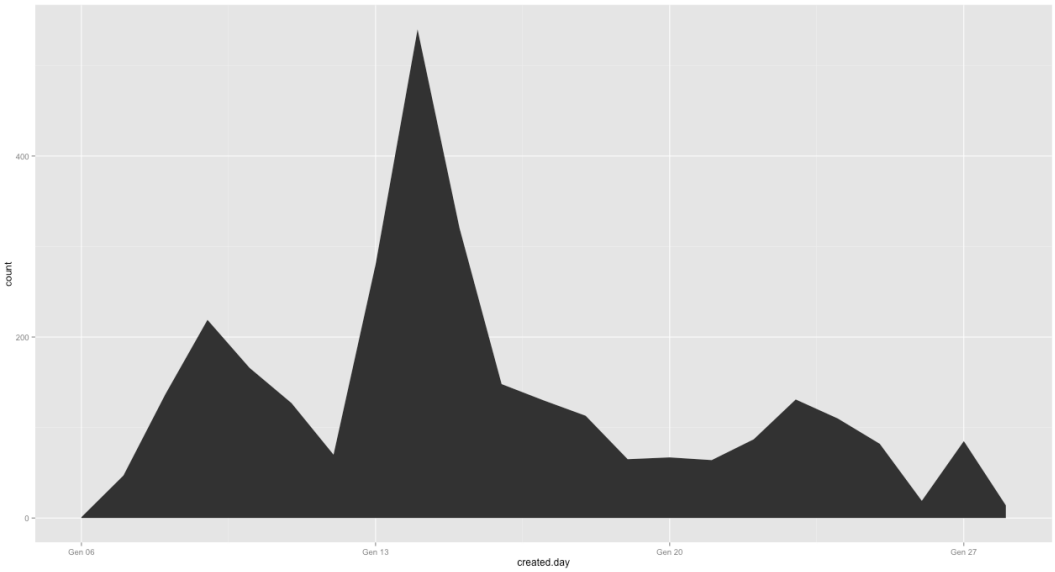
```
user.date=cbind(tw$screenName,tw$created.day)
sort(table(user.date[,1]),decreasing=T)[1:20]
```

Lo user più attivo sembra essere *orianoPER*, osserviamo ora la distribuzione dei tweet nel tempo in figura 1.2, avvalendoci dei seguenti comandi:

```
orianoPER=subset(user.date,user.date[,1]=="orianoPER")
plot(table(orianoPER[,2]))
```



(a) Distribuzione oraria



(b) Distribuzione giornaliera

Figura 1.1: Distribuzioni rispettivamente orarie e giornaliere dei tweet

	Conteggi		Conteggi
orianoPER	103	CiriacoPia	16
Animalisti_FVG	69	LAVonlus	15
Baby970	54	salatina67	15
ilbrescio	36	gravitazeroeu	14
clizia72ita	22	MauraBracaloni	14
IMorsanutto	22	TerrinoniL	14
RosselladiKira	22	GraziaIotti	13
scinet_it	21	cocopress	12
danielebanfi83	19	AleCusinato	11
Alex_Colla	17	eugeniosantoro	11

Tabella 1.3: Conteggi degli users piú attivi

1.4 La Document Term Matrix

Costruiamo la Document Term Matrix, che verrà utilizzata nelle prossime analisi. Questa matrice descrive le frequenze dei termini presenti in una collezione di documenti: le righe corrispondono ai documenti e le colonne ai termini. Vediamo una sezione di esempio presa dalla matrice che sará utilizzata in seguito (tabella 1.4).

	basta	bioetica	bisogna	caterina	cerchiamo	ci	come	conosce...
45	0	0	0	1	0	0	0	0
46	0	1	0	0	0	0	0	0
47	0	0	0	0	0	1	0	0
48	0	0	0	0	0	0	0	0
49	0	0	1	0	0	0	0	1
50	0	1	0	0	0	0	2	0

Tabella 1.4: Estratto di DTM

Il comando `DocumentTermMatrix` necessita di argomenti di tipo vector, per prima cosa quindi va creato il vettore di testi (in questa sezione ci appoggiamo alla libreria `tm.plugin.webmining`(Annau, 2012)). Diamo una breve descrizione delle opzioni del comando:

1. Eliminiamo lo stemming, il processo di riduzione della forma flessa di una parola alla sua forma radice, detta tema

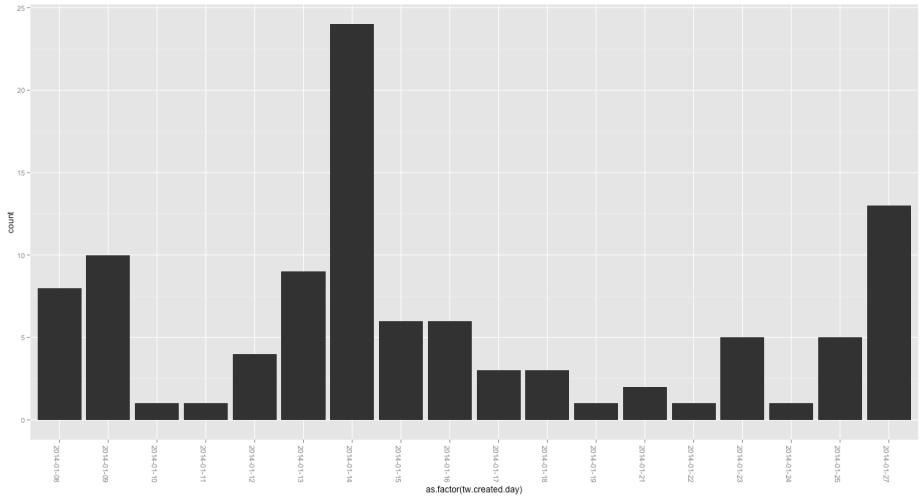


Figura 1.2: Distribuzione nel periodo considerato dei tweet relativi all'utente orianoPER

2. Scegliamo come insieme di stopword quelle italiane presenti nella libreria TextWiller (quelle parole che, per la loro alta frequenza in una lingua, sono di solito ritenute poco significative dai motori: gli articoli, le preposizioni e le congiunzioni...)(La lista é presentata in appendice)

3. Consideriamo come lunghezza minima una parola composta da due lettere

4. Infine scegliamo di non rimuovere numeri e punteggiatura

```
corpus=Corpus(VectorSource(tw$text))

dtm=DocumentTermMatrix(corpus,
            control = list( stemming = FALSE,
            stopwords=itastopwords, minWordLength = 2,
            removeNumbers = FALSE, removePunctuation = FALSE,
            bounds=list(local = c(1,Inf)) ))
```

1.5 Analisi degli N-grammi

Introduciamo ora un concetto utile per proseguire le analisi, l'*n-gramma*. Un n-gramma è una sottosequenza di n elementi di una data sequenza; in

base all'applicazione, gli elementi in questione possono essere fonemi, sillabe, lettere, parole, ecc. Concentriamoci inizialmente nella ricerca di parole e di n-grammi più frequenti, allo scopo di carpire quali sono gli argomenti più discussi. Per estrarre gli n-grammi più utilizzati ci appoggiamo alla funzione `textcnt` del pacchetto `tau` (Buchta, Hornik, Feinerer, and Meyer, 2012); gli esempi seguenti si limitano a prendere in considerazione bigrammi e trigrammi, ordinati per frequenza: è possibile ovviamente variarne la lunghezza, ma trattandosi di tweet, quindi di frasi molto brevi, aumentare le parole porta a risultati di poco interesse.

```
bigrams=textcnt(tw$text,method="string",n=2L,split="[[:blank:]]")
sort(bigrams,decreasing=TRUE)[1:15]
```

	Conteggi		Conteggi
sperimentazione animale	701	rt @animalisti_fvg:	73
la sperimentazione	380	con la	72
sperimentazione animale,	268	sperimentazione animale	71
sulla sperimentazione	242	#vivisezione non	67
rt @orianoper:	238	NA NA	65
sperimentazione animale:	228	italiano non	65
sperimentazione animale.	188	la #sperimentazione	61
alla sperimentazione	161	il governo	59
della sperimentazione	155	governo italiano	59
animale e	111	contro la	56
per la	102	stiamo con	56
rt @lavonlus:	98	al senato	55
#sperimentazione animale	86	nasce un	52
fermi la	78	non é	52
non fermi	77	su sperimentazione	52

Tabella 1.5: Bigrammi individuati

```
trigrams=textcnt(tw$text,method="string",n=3L,split="[[:blank:]]")
sort(bigrams,decreasing=TRUE)[1:30]
```

Una volta individuati i bigrammi e i trigrammi più rilevanti, possiamo sfruttarli nell'analisi sostituendoli nei tweet come parola unica: in questo modo "sperimentazione animale" diventerà "sperimentazione_animale". Evitiamo dunque che i conteggi delle parole più usate (sperimentazione, animale...) aumentino vertiginosamente, facendo diminuire inevitabilmente l'importanza di altre. Di seguito sono presentati i comandi e tutte le sostituzioni fatte:

	Conteggi		Conteggi
sperimentazione animale sperimentazione	93	animale sperimentazione peluche	38
fermi sperimentazione animale	78	brambilla basta sperimentazione	35
governo italiano fermi	65	sperimentazione animale ecco	35
italiano fermi sperimentazione	65	sperimentazione animale senato	35
cattaneo governo italiano	56	favore sperimentazione animale	34
elena cattaneo governo	55	vivisezione risulta nocivo	32
basta sperimentazione animale	52	animale insidie web	31
ruolo sperimentazione animale	50	risulta nocivo spiegatelo	31
animale sperimentazione animale	49	sperimentazione animale insidie	31
nasce farmaco ruolo	49	sperimentazione animale vivisezione	31
farmaco ruolo sperimentazione	48	legge sperimentazione animale	30
vivisezione sperimentazione animale	48	sperimentazione animale diritto	29
senato sperimentazione animale	42	animale diritto conoscenza	28
convegno sperimentazione animale	40	diritto conoscenza salute	28
parla sperimentazione animale	40	caterina dimostriamolo adesso	27

Tabella 1.6: Trigrammi individuati

```
gsub("sperimentazione animale","sperimentazione_animale",tw$testo)
gsub("sperimentazione  animale","sperimentazione_animale",testi)
gsub("sperimentazioneanimale","sperimentazione_animale",testi)
gsub("governo italiano","governo_italiano",testi)
gsub("non fermi","non_fermi",testi)
```

1.6 Un po' di grafici

Abbiamo finora esposto il procedimento per alcune semplici analisi descrittive riguardanti la composizione del nostro dataset. Veniamo ora alla parte grafica, soffermandoci in particolare sulla costruzione di una wordcloud e delle reti di parole e retweet.
Come appena accennato é possibile costruire, con i termini più utilizzati, una wordcloud: una rappresentazione visiva delle etichette o parole-chiave. Il peso delle etichette che viene reso con caratteri di dimensioni diverse è inteso esclusivamente come frequenza di utilizzo; più grande il carattere, maggiore la frequenza della parola (figura 1.3). La libreria usata é `wordcloud`(Fellows, 2013), con i seguenti comandi:

```
require(wordcloud)
wordcloud(words=colnames(dtm2),freq=colSums(as.matrix(dtm)),
        min.freq=40,color="darkred")
```

Figura 1.3: Wordcloud basata sulla Document Term Matrix creata in precedenza

Facciamo un passo avanti, e vediamo come costruire un grafo che rappresenti le connessioni tra le parole piú frequenti e le connessioni tra gli utenti che si generano attraverso il processo di retweet (i seguenti grafici sono disponibili con il pacchetto `igraph`(Csardi and Nepusz, 2006)).
Iniziamo con il grafo delle parole (figura 1.4), soffermandoci passo passo sulle fasi del procedimento.

```
wc = rowSums(t(as.matrix(dtm)))
m=t(as.matrix(dtm))

lim = quantile(wc, probs=0.99)
good = m[wc > lim,]
```

Partiamo con i conteggi delle parole, sfruttando la Document Term Matrix, e prendiamone il sottoinsieme con frequenza maggiore allo 0.99 (la scelta é dettata da un semplice fattore estetico, aumentando troppo il numero di termini il grafo che ne risulta é poco leggibile).

```
good = good[,colSums(good)!=0]
good=subset(good,row.names(good)!=c("suppressedtext","dettagli:"))

M = good %*% t(good)
diag(M) = 0
```

Ora che abbiamo ottenuto le parole, eliminiamo quelle che presentano degli zeri nelle colonne, e calcoliamo la matrice di adiacenza: una matrice binaria quadrata che ha come indici di righe e colonne i nomi dei vertici del grafo. Nel posto (i,j) della matrice si trova un 1 se e solo se esiste nel grafo un arco che va dal vertice i al vertice j, altrimenti si trova uno 0. Due comandi non sono stati commentati, quelli con `subset`: anche questa é stata una scelta arbitraria, il termine *suppressedtext* non ha alcuna valenza mentre *dettagli:* risultava fuorviante nella rappresentazione grafica.

```
g = graph.adjacency(M, weighted=TRUE, mode="undirected",
                    add.rownames=TRUE)

glay = layout.fruchterman.reingold(g)

kmg = kmeans(M, centers=8)
```

```
gk = kmg$cluster

gbrew = c("red", brewer.pal(8, "Dark2"))
gpal = rgb2hsv(col2rgb(gbrew))
gcols = rep("", length(gk))
for (k in 1:8) {
  gcols[gk == k] = hsv(gpal[1,k], gpal[2,k], gpal[3,k], alpha=0.5)
}

V(g)$size = 5
V(g)$label = V(g)$name
V(g)$degree = degree(g)
V(g)$label.cex = 1 * log10(V(g)$degree)
V(g)$label.color = hsv(0, 0.1, 0.5, 0.8)
V(g)$frame.color = NA
V(g)$color = gcols
E(g)$color = hsv(0, 0, 0.3, 0.2)

plot(g, layout=glay)
```

Questi ultimi comandi riguardano la parte puramente estetica, per darne una breve descrizione:

1. `graph.adjacency` e `layout.fruchterman.reingold` creano la struttura del grafico che servirá al comando finale
2. `kmeans` crea una clusterizzazione dei termini con il metodo delle k-medie per la scelta dei colori
3. il ciclo `for` assegna i colori ai vertici
4. la lista di comandi `V(g)` modificano le opzioni di visualizzazione
5. il comando finale `plot` ci porge finalmente l'output

Le analisi del prossimo capitolo saranno basate prevalentemente sui retweet, come preannunciato vediamo allora come creare un grafo che colleghi gli users tra loro (figura 1.5).

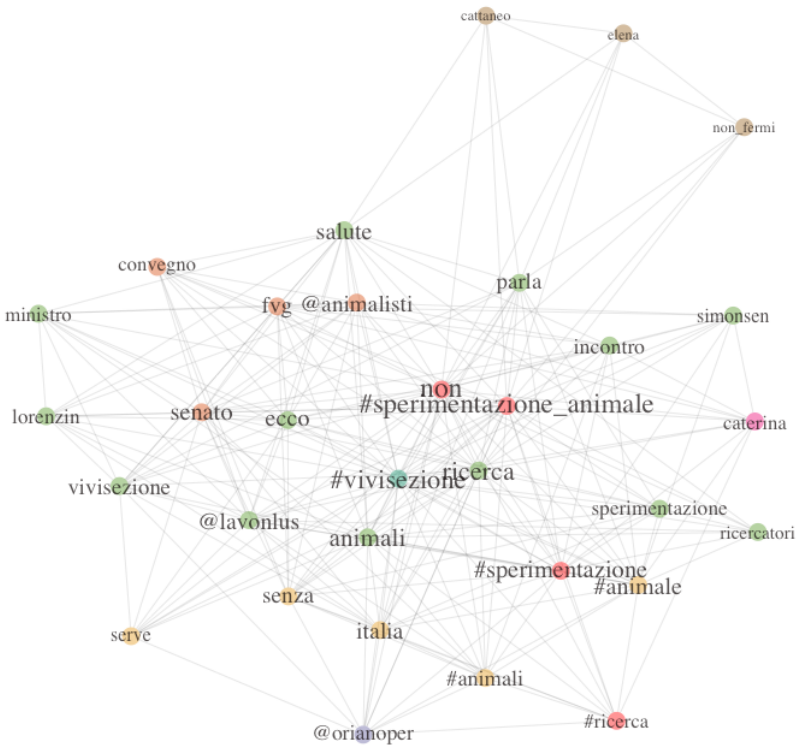


Figura 1.4: Grafo delle parole che co-occorrono piú frequentemente

```
trim=function (x) sub('@','',x)

tw$to=sapply(tw$text,function(tweet)
        trim(str_extract(tweet,"^(@[[:alnum:]_]*)")))
tw$rt=sapply(tw$text,function(tweet)
        trim(str_match(tweet,"^RT (@[[:alnum:]_]*)")[2]))

ats.df=data.frame(tw$screenName,tw$to)
rts.df=data.frame(tw$screenName,tw$rt)

ats.g=graph.data.frame(subset(ats.df,!is.na(tw.to), directed=T)
rts.g=graph.data.frame(subset(rts.df,!is.na(tw.rt)), directed=T)

plot(ats.g,vertex.label.color=hsv(h=0, s=0, v=.95, alpha=0.5),
     vertex.label.cex=0.5,vertex.size=5,edge.arrow.size=0.2,
     edge.arrow.width=0.2, edge.width=1)
plot(rts.g,vertex.label.color=hsv(h=0, s=0, v=.95, alpha=0.5),
     vertex.label.cex=0.5,vertex.size=5,edge.arrow.size=0.2,
     edge.arrow.width=0.2, edge.width=1)
```

Spieghiamo meglio i passaggi:

1. Definiamo la funzione `trim`,che elimina i caratteri
2. Creiamo le variabili `to` e `rt` per identificare gli user
3. Costruiamo i dataframe associando le variabili appena create con lo username cui appartiene il tweet e con il comando `graph.data.frame` la struttura del grafo
4. Come sempre, `plot` é il passaggio finale

Questo breve excursus ci ha permesso di investigare in superficie il dataset che abbiamo a disposizione, ma le strade per approfondire la nostra conoscenza sull'argomento sono ancora molteplici. Nei capitoli che seguono diamo un assaggio di come estrapolare quanta piú informazione possibile dai nostri dati testuali.

Figura 1.5: Rete di retweet tra gli autori

Capitolo 2

Analisi dei retweet tramite RTHound

Modifichiamo l'approccio ai tweet, basandoci non più su singole parole (o n-grammi), ma sull'intero messaggio: si cerca quindi di individuare, se esiste, la presenza di tweet predominanti, contenenti le opinioni più significative presenti nel dataset in analisi. L'assunto fondamentale é l'importanza che hanno i retweet: un tweet che venga utilizzato piú volte dallo stesso o da altri utenti assume rilevanza nella nostra analisi, in quanto indica che l'informazione in esso contenuta ha grande valenza nell'argomento di discussione. Il problema consiste peró nell'individuare i retweet: i dataset ottenibili con le procedure introdotte all'inizio del capitolo non sempre ci permettono di estrarli facilmente. I motivi principali sono due: o Twitter non é riuscito a codificare il messaggio come retweet, o l'utente ha citato parzialmente o apportato modifiche al testo. Durante lo svolgimento di tutte le analisi si é sviluppata una possibile idea per ovviare al problema.

2.1 La funzione RTHound

L'idea consiste nello sfruttare una apposita funzione di distanza tra testi: quella usata è la *distanza di Levenshtein*, o distanza di edit. La distanza di Levenshtein tra due stringhe A e B è il numero minimo di modifiche elementari che consentono di trasformare A in B; per modifiche elementari si intende la cancellazione di un carattere, la sostituzione di un carattere con un altro, l'inserimento di un carattere. L'algoritmo usato comunemente per calcolare la distanza richiede l'uso di una matrice $(n+1) \times (m+1)$, con n e m rappresentanti le lunghezze delle due stringhe.

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{se } \min(i,j) = 0 \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j) + 1 \\ \mathrm{lev}_{a,b}(i,j-1) + 1 \\ \mathrm{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{altrimenti} \end{cases}$$

Per fare un esempio:

```
levenshteinDist("cane",c("cono","coro"))
```

```
[1] 2 3
```

Si è costruita quindi una funzione ad hoc in R, che costruisce una matrice quadrata contenente le distanze tra tutte le possibili combinazioni di coppie di tweets. Questa matrice servirà per clusterizzare i dati, accorpando i più vicini (simili nel nostro caso) tra loro; una volta completata la procedura si cercherà il tweet più vecchio in ordine temporale in ogni cluster, che viene considerato il tweet originale e riassumerá quindi l'informazione contenuta in quel gruppo. La funzione scritta appositamente, che incorpora le varie fasi dell'analisi appena descritta, è `RTHound`, anch'essa contenuta nel pacchetto `TextWiller`.Per ovviare al sovraccarico computazione, la funzione divide il dataset di testi ordinati temporalmente in sottoinsiemi di cardinalità prefissata, aggiungendo inoltre dal secondo subset un numero dato di tweets del precedente; calcola la matrice di dissimilarità basata sulla *distanza di Levenshtein* per ogni subset e raggruppa i tweets attraverso un algoritmo di clusterizzazione gerarchica con metodo *"complete"*. Infine i tweets appartenenti allo stesso cluster vengono rimpiazzati dal tweet più vecchio, identificandoli come retweets; a questo punto è possibile attraverso un conteggio ordinare retweet

più frequenti. Il dataset viene diviso in subset per motivi computazionali: per evitare perdita di informazioni causate da questa procedura, si è deciso di incorporare in ogni subset (tranne il primo), un certo numero di tweet del precedente. Questo perchè si suppone che i retweet abbiamo un periodo di vita limitato, in genere un tweet nell'arco di una settimana completa il suo ciclo di retweet. Esiste anche un'altra limitazione, è possibile che alcuni dei tweet principali (quelli da cui sono partiti i retweet), non siano stati raccolti; per ovviare al problema, prendiamo i tweet più vecchi per ogni cluster, siano essi tweet o già retweet.

2.2 Come funziona

Vediamo in dettaglio come lavora la funzione. I parametri che la caratterizzano sono tre:

1. *testo* é il vettore di tweet
2. S rappresenta la cardinalitá di ogni sottoinsieme
3. L il numero di tweet da accorpare ad ogni sottoinsieme (escluso il primo)
4. *hclust.dist* l'altezza corrispondente al taglio dell'albero
5. *hclust.method* il metodo di clustering
6. *showTopN* il numero di retweet da mostrare come output

```
function (testo, S = 500, L = 100, hclust.dist = 100,
        hclust.method = "complete", showTopN = 5)
{
    testo.na = which(is.na(testo))
    ntesti = length(testo)
    if (length(testo.na) > 0)
        testo = testo[-testo.na]
    nWindows = (floor(length(testo)/S) - 1)
    s = c(0:nWindows)
```

Questa prima parte estrae il sottoinsieme di tweet effettivi, eliminando quelli senza testo. Si inizializzano inoltre le variabili $nWindows$ ed s che serviranno per determinare i subset di tweet per ogni ciclo.

```
for (l in 1:length(s)) {
      cat("\nWindow #", l)
      if (l != length(s)) {
          ids = c(((S) * s[l]):((S) * s[l + 1]))
          select = testo[ids]
      }
      else {
          select = testo[((S) * s[l]):length(testo)]
      }
      if (l > 1) {
          selectPeriodoPrima = testo[((S) * s[l] -
          (L + 1)):((S) * s[l] - 1)]
          select = c(selectPeriodoPrima, select)
      }
```

Il ciclo for applica i comandi per ogni sottoinsieme di tweet, i due comandi if ci permettono di distinguere il primo sottoinsieme in ordine temporale dagli altri: come giá detto infatti dal secondo subset vengono accorpati L tweet da quello precedente.

```
m = matrix(ncol = length(select), nrow = length(select))
      for (i in 1:(length(select) - 1)) {
          for (j in (i + 1):length(select)) {
              m[i, j] = levenshteinDist(testo[i], testo[j])
          }
      }
      m = as.dist(t(m))
      h = hclust(dist(m), method = hclust.method)
      tree = cutree(h, h = hclust.dist)
```

Questo é il fulcro dell'intera funzione, dove viene costruita la matrice di distanze m tra tweet con la distanza di Levensthein: da essa si identificano tramite clusterizzazione gerarchica i gruppi con il comando `hclust`, selezionati tagliando l'albero risultante ad una altezza variabile h con il comando `cutree`.

```
   idClusters = sapply(unique(tree), function(x)
   which(tree == x))
      for (i in 1:length(idClusters))
```

```
        testo[idClusters[[i]]] = testo[idClusters[[i]][1]]
    }
    if (showTopN > 0)
        cat("\n", showTopN, " most frequent RTs\n",
        sort(table(testo), decreasing = T)[1:showTopN])
    if (length(testo.na) > 0) {
        testoOut = rep("", ntesti)
        testoOut[-testo.na] = testo
        testo = testoOut
    }
    testo
}
```

Quest'ultima parte si occupa di identificare i cluster di tweet, e sostituisce i tweet di ogni cluster con il tweet (o retweet) più vecchio. Non ci resta che applicarla al nostro dataset:

```
RTHound(tw$text)
```

	Conteggi	ID
RT @lercionotizie: Brambilla: Basta con la sperimentazione animale, sí alla sperimentazione sui peluche	25	top1
RT @orianoPER: Questa foto é stata scattata in Italia.GreenHill. Questo era un modello #animale destinato a #sperimentazioneanimale http:/?	19	top2
RT @LAVonlus: Sperimentazione animale,ecco perché non andremo a Convegno Senato:lettera a @PietroGrasso e Senatori #opensenato http://t.co/?	16	top3
RT @LAVonlus: Tre miti da sfatare sulla #vivisezione. Le risposte ai luoghi comuni della sperimentazione animale http://t.co/zHSfam16DT	16	top4
RT @orianoPER: La #sperimentazioneanimale é inutile perché non predittiva per la specie umana. MEDICI ANTI #VIVISEZIONE- LIMAV http://t.co/?	15	top5

Tabella 2.1: Retweet individuati da RTHound ordinati per frequenza

2.3 Commenti

I retweet piú rilevanti sono stati individuati, e con essi un riassunto delle idee predominanti contenute nei tweet del dataset. La scelta di prendere i primi cinque é stata fatta sulla base dei conteggi: l'inclusione di altri non avrebbe portato ad un gran guadagno di informazione. Si é visto nel corso dell'analisi che anche la scelta della variabile `h,` oltre a quelle giá incluse nella funzione, influisce sulla grandezza dei conteggi; il rischio peró é quello di accorpare tweet molto diversi tra loro o, dal lato opposto, non creare clusterizzazione. Una ulteriore difficoltá é scegliere una funzione di distanza adeguata per i testi che abbiamo. Ovviamente la distanza di Levenshtein non é la sola, ma in questo caso sembra essere la piú aderente alle ipotesi.
Come detto all'inizio dell'analisi, i retweet hanno un tempo di vita molto breve, con l'aiuto del grafico della distribuzione per giorni (figura 2.1) vediamo che non superano i due o tre giorni.

La funzione RTHound ci tornerá utile anche nel prossimo capitolo, in particolare l'aver sostituito i tweet appartenenti ad uno stesso cluster con il piú vecchio. Questo ci permette, con l'aiuto della funzione `unique,`di eliminare tutti i retweet individuati. Il motivo é semplice: se il nostro scopo é di individuare un insieme di topic, la presenza di retweet distorce i nostri risultati. Se i retweet sono presenti piú volte, i modelli tenderanno ad identificarli come topic: questo ovviamente non ci aiuta, noi siamo interessati agli argomenti trattati in tutto il corpus, non all'argomento specifico di un tweet. Vedremo quindi nel prossimo capitolo come sfruttare questa possibbilitá.

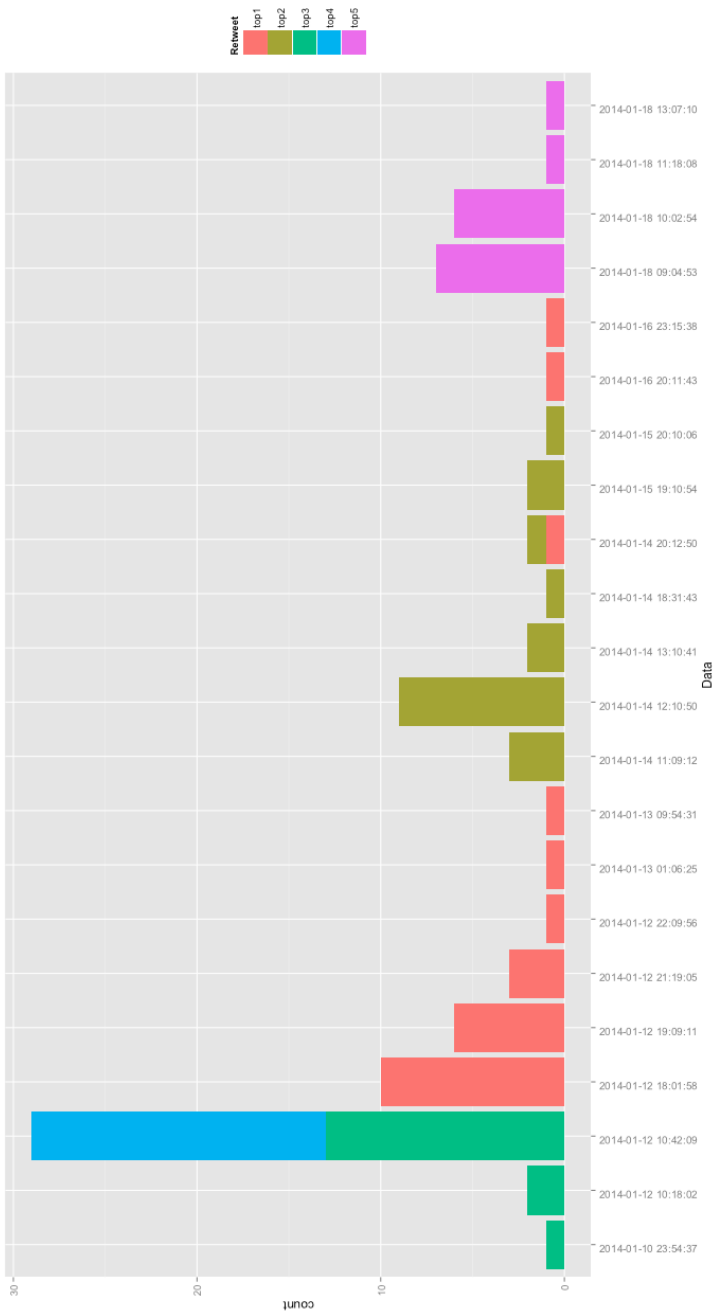


Figura 2.1: Distribuzione dei retweet individuati. I conteggi sono individuabili dai diversi colori (tabella 2.1), ordinati come proposto sopra

Capitolo 3

Topic Model Analysis su dati di microblogging

Si è visto nella prima parte come l'analisi sui tweets sia difficoltosa e porti a risultati non sempre facilmente interpretabili. In questa seconda parte si vogliono analizzare testi più complessi e corposi, comprensivi di più argomenti, ma mantenendo il filo conduttore della sperimentazione animale. I metodi utilizzati sono diversi dai precedenti e lo scopo è anche quello di confrontare tra loro gli ultimi sviluppi dei topic model. I topic model vengono utilizzati per analizzare grandi quantità di informazioni testuali allo scopo di studiare la modellazione del linguaggio, la classificazione di documenti, la sintetizzazione di documenti e non ultimo il data mining. Più specificamente, dato un insieme di documenti, la stima dei parametri in questi modelli estrae un insieme ristretto di distribuzioni sulle parole, definite *topic*.

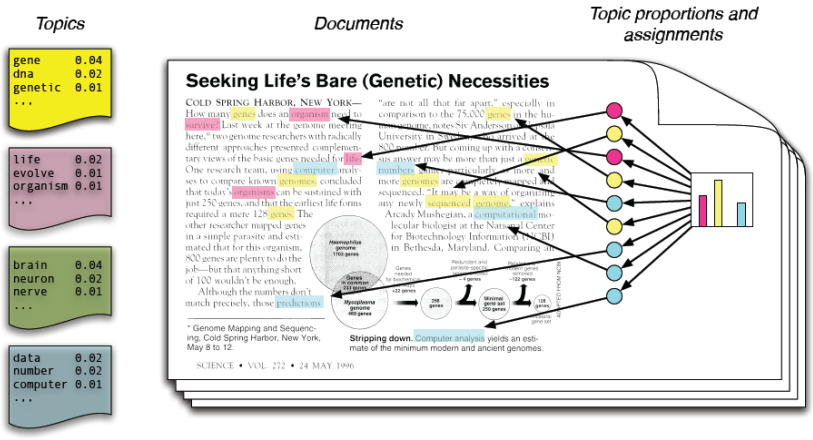


Figura 3.1: Esempio introduttivo di topic model

3.1 Modello teorico

Come illustrato in figura 3.1 un generico testo è costituito da una mistura di argomenti, z, a cui sono associate parole specifiche di un vocabolario; ognuno di questi argomenti ha una relativa distribuzione di probabilità sulle parole del vocabolario: pensiamo ad esempio all'argomento sperimentazione, avrà sicuramente un linguaggio diverso dall'argomento giardinaggio. Ovviamente il topic model non è in grado di dirci quale sia l'argomento, ma attraverso le distribuzioni sulle parole otterremo dei cluster di termini, ove ognuno di questi cluster corrisponde ad un topic a cui saranno associati i termini con probabilità più alta (colonna sinistra in figura 3.1). Il processo di generazione di un documento segue dunque queste fasi: si sceglie un topic, un argomento, e condizionatamente al topic scelto si estrae una parola; iterando il procedimento si arriva a costruire un intero testo. Certo la mente umana non procede in questo modo, il processo generativo infatti andrebbe visto a posteriori, ed è proprio a quel punto che entra in gioco il modello statistico.

L'assunto di base per questi modelli è che i testi siano composti da una mistura di argomenti, topic, aventi una possibile correlazione tra loro; ognuno di questi topic è una distribuzione multinomiale sulle parole, queste ultime raggruppate in un vocabolario definito in precedenza sulla base dei testi analizzati: le parole con probabilitá più alta forniscono un'idea dei temi trattati nella collezione di documenti. Un topic model è dunque un modello per la generazione di documenti: per generare un nuovo testo si estrae una distri-

buzione sui topic, quindi, per ogni parola, si sceglie un topic casuale e si estrae una parola da questa distribuzione. Ovviamente il processo può essere invertito attraverso tecniche statistiche, allo scopo di fare inferenza sull'insieme di topic che ha generato il documento. Sono stati proposti svariati modelli per l'analisi dell'informazione contenuta nei documenti e del significato delle parole; questi hanno in comune un presupposto fondamentale, un documento è una mistura di topic, come accennato in precedenza, e si differenziano per assunzioni statistiche. Per introdurre la notazione, indichiamo con $\Pr(z)$ la distribuzione di probabilità sui topic in un particolare documento, con $\Pr(w|z)$ la distribuzione di probabilità dato un topic z; $\Pr(z_i = j)$ sarà la probabilità che il j-esimo topic sia estratto per la i-esima parola e $\Pr(w_i|z_i = j)$ la probabilità della parola w_i sotto il topic j. Il modello deriva la seguente distribuzione sulle parole in un documento:

$$\Pr(w_i) = \sum_{j=1} \Pr(w|z = j) \Pr(z = j)$$

La formula appena presentata descrive il più generico topic model e fornisce uno spunto di base per comprenderne il funzionamento. L'approccio utilizzato é di tipo bayesiano: per stimare la probabilità di trovare una parola in un testo ci basiamo sul prodotto tra la probabilità di trovare un certo topic, P(z), e la probabilità di trovare la stessa parola condizionatamente al topic scelto, P(w|z).

3.2 Interpretazione geometrica

Il topic model ha un'elegante interpretazione geometrica. Dato un vocabolario contenente W parole distinte, esso definisce uno spazio W dimensionale dove ogni asse corrisponde alla probabilità di osservare una specifica parola. Il simplesso $W - 1$ dimensionale identificato rappresenta tutte le distribuzioni di probabilità sulle parole. In figura 3.2 la regione ombreggiata corrisponde al simplesso bidimensionale che rappresenta tutte le ditribuzioni di probabilità sulle tre parole. Come distribuzione di probabilità sulle parole, ogni documento può essere identificato da un punto sul simplesso; allo stesso modo, ogni topic può essere identificato da un punto sul simplesso. Ogni documento che viene generato dal modello è una combinazione convessa dei T topic che non solo identifica tutte le ditribuzioni di parole come punti sul

simplesso $W-1$ dimensionale, ma anche come punti del simplesso $T-1$ dimensionale generato dai topic. A questo punto dell'analisi ci si può chiedere

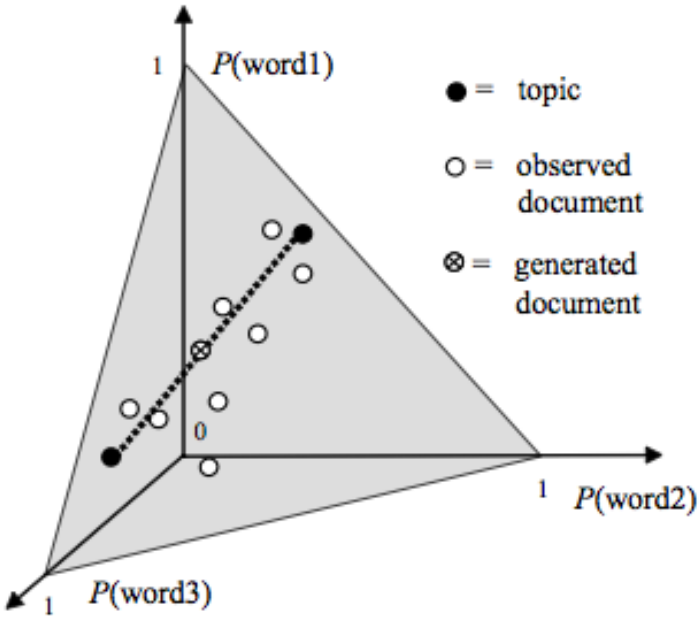


Figura 3.2: Interpretazione geometrica del topic model

come individuare quali siano i documenti più simili, o scovare i documenti più attinenti data una qualche query: un nuovo insieme di parole ad esempio o un insieme di parole già esistenti nella collezione di documenti. Definiamo prima di tutto il concetto di somiglianza tra documenti, molto simile ai concetti usati nella prima parte sulla somiglianza tra tweets. La somiglianza tra documenti può essere misurata attraverso la corrispettiva somiglianza tra distribuzioni di probabilità dei topic. Esistono varie alternative di funzioni di somiglianza per distribuzioni di probabilità. Una delle funzioni prese in esame è la *distanza di Kullback-Leibler* (anche detta divergenza di informazione, entropia relativa, o KLIC): è una misura non simmetrica della differenza tra due distribuzioni di probabilità P e Q. Specificamente, la divergenza di Kullback-Leibler di Q da P, indicata con $DKL(P||Q)$, è la misura dell'informazione persa quando Q è usata per approssimare P. Tipicamente P rappresenta la vera distribuzione di dati, osservazioni, o una distribuzione teorica calcolata con precisione. La misura Q tipicamente rappresenta una teoria, modello, descrizione, o approssimazione di P. Anche se è spesso pen-

sata come una distanza, la divergenza KL non è una vera e propria metrica - per esempio, non è simmetrica: la KL da P a Q non è in genere la stessa KL da Q a P. Data la non simmetria, può essere conveniente utilizzare una forma simmetrica:

$$KL(p,q) = \frac{1}{2}\left(D(p,q) + D(q,p)\right)$$

Un'altra opzione consiste nell'applicare la *distanza di Jensen-Shannon*:

$$JS(p,q) = \frac{1}{2}\left(D\left(p, \frac{p+q}{2}\right) + D\left(q, \frac{p+q}{2}\right)\right)$$

è una funzione simmetrica che misura la somiglianza tra p e q attraverso la loro media: due distribuzioni p,q sono simili se sono simili alla loro media $\frac{p+q}{2}$. Entrambe le versioni simmetriche KL e JS lavorano bene nei dati reali. E' inoltre possibile considerare le distribuzioni di probabilità dei topic come vettori e applicare geometricamente la distanza Euclidea, il prodotto interno o il coseno. I due metodi appena proposti si limitano a calcolare una misura per la somiglianza tra documenti; piú utile sarebbe un approccio probabilistico.

3.3 Latent Dirichlet Allocation (LDA)

Come appena accennato, abbiamo bisogno di un approccio probabilistico. Si cerca quel documento che massimizza la probabilità condizionata della query: la stima dei parametri dei modelli discussi in seguito si basa proprio su questa formula. Definendo $\Pr(q|d_i)$, con q insieme di parole della query, e considerando le usuali assunzioni dei topic model, si calcola:

$$\Pr(q|d_i) = \prod_{w_k \in q} \Pr(w_k|d_i) = \prod_{w_k \in q} \sum_{j=1}^{T} \Pr(w|z=j)\Pr(z=j|d)$$

Da notare che l'approccio appena descitto enfatizza la somiglianza tra topic, individuando come documenti più rilevanti quelli che hanno distribuzione di probabilità dei topic più vicina possibile all'insieme di parole associate alla query. Il modello LDA, *Latent Dirichlet allocation* (Blei, M., Ng, Y., Jordan, and I., 2003), è un topic model usato comunemente: esso rappresenta ogni documento come una mistura di topic, ove ogni topic è una distribuzione multinomiale sulle parole del vocabolario. Per la generazione di un documento si procede come segue:

1. Si estrae $\theta_i \sim \text{Dir}(\alpha)$, dove $i \in \{1, \ldots, M\}$ e $\text{Dir}(\alpha)$ è la distribuzione di *Dirichlet* per il parametro α
2. Si estrae $\varphi_k \sim \text{Dir}(\beta)$, dove $k \in \{1, \ldots, K\}$
3. Per ogni valore i , j della parola, dove $j \in \{1, \ldots, N_i\}$, e i $\in \{1, \ldots, M\}$
   (a) Si estrae un topic da $z_{i,j} \sim \text{Multinomiale}(\theta_i)$
   (b) Si estrae una parola da $w_{i,j} \sim \text{Multinomiale}(\varphi_{z_{i,j}})$

Il modello finale per le parole è:

$$\Pr(\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{\varphi}; \alpha, \beta) = \prod_{i=1}^{K} \Pr(\varphi_i; \beta) \prod_{j=1}^{M} \Pr(\theta_j; \alpha) \prod_{t=1}^{N} \Pr(Z_{j,t}|\theta_j) \Pr(W_{j,t}|\varphi_{Z_{j,t}})$$

Proviamo subito ad eseguire un'analisi LDA, partendo direttamente dal dataset di tweets; ci concentreremo più avanti sui testi. Per prima cosa come sempre normalizziamo il testo dei tweet unici (come giá spiegato alla fine del capitolo precedente):

```
corpus=normalizzaTesti(unique(tw$text),normalizzacaratteri=TRUE,
tolower=TRUE,perl=FALSE,fixed=FALSE)
```

Ora facciamo la lessicalizzazione del corpus e successivamente scegliamo il vocabolario di parole da utilizzare (ci appoggiamo ad una variabile corpus1 per evitare problemi che potrebbe dare il comando lexicalize se utilizzato due volte sullo stesso oggetto):

```
corpus1=lexicalize(corpus$testo)
to.keep.voc=corpus1$vocab[word.counts(corpus1$documents,
        corpus1$vocab) >= 3]
to.keep.stop=subset(to.keep.voc,is.na(pmatch
        (to.keep.voc,itastopwords)))
corpus=lexicalize(corpus$testo,vocab=to.keep.stop)
```

Il vettore `itasopwords` è presente nella libreria già citata `TextWiller`. Inizializziamo le variabili che ci serviranno: N é il numero di righe del nostro dataset, quindi il numero di tweet, K il numero di cluster, Top la variabile che indica il numero di termini di cui sará composto ogni cluster ed infine I il numero di iterazione. La scelta di prendere 10 cluster é voluta; secondo

il criterio AIC il numero di cluster dovrebbe essere 3, ma il nostro scopo é ottenere quanta piú informazione possibile sugli argomenti trattati, e sarebbe controproducente limitarsi ad un numero cosí ristretto.

```
L         K       AIC
-56642.60 2  113289.2
-56602.31 3  113210.6
-56883.29 4  113774.6
-56955.50 5  113921.0
-58396.15 10 116812.3
-62189.24 20 124418.5
-64925.82 30 129911.6
-70822.17 50 141744.3
```

Possimo a questo punto lanciare il comando `lda.collapsed.gibbs.sampler` (la libreria di riferimento é lda (Chang, 2012)).

```
N  = nrow(corpus)
K  = 10
Top = 10
I = 50
result = lda.collapsed.gibbs.sampler(corpus, K, to.keep.stop,
        I,  0.1, 0.1, compute.log.likelihood=TRUE)
```

La funzione ha calcolato i 10 topic da noi richiesti; prima di analizzarli, controlliamo lo stabilizzarsi della funzione di verosimiglianza con i seguenti comandi (figura 3.3):

```
plot(c(1:I),result$log.likelihoods[1,],type="l",ylim=c(min(min
        (result$log.likelihoods[1,]),
        min(result$log.likelihoods[2,])),
        max(max(result$log.likelihoods[1,]),
        max(result$log.likelihoods[2,]))),
        col="blue",xlab="Iterations",ylab="LogLikelihood")
lines(result$log.likelihoods[2,],col="red")
```

Appurato lo stabilizzarsi della log-verosimiglianza, osserviamo la composizione dei topic in tabella 3.1:

```
top.words=top.topic.words(result$topics, Top, by.score=TRUE)
```

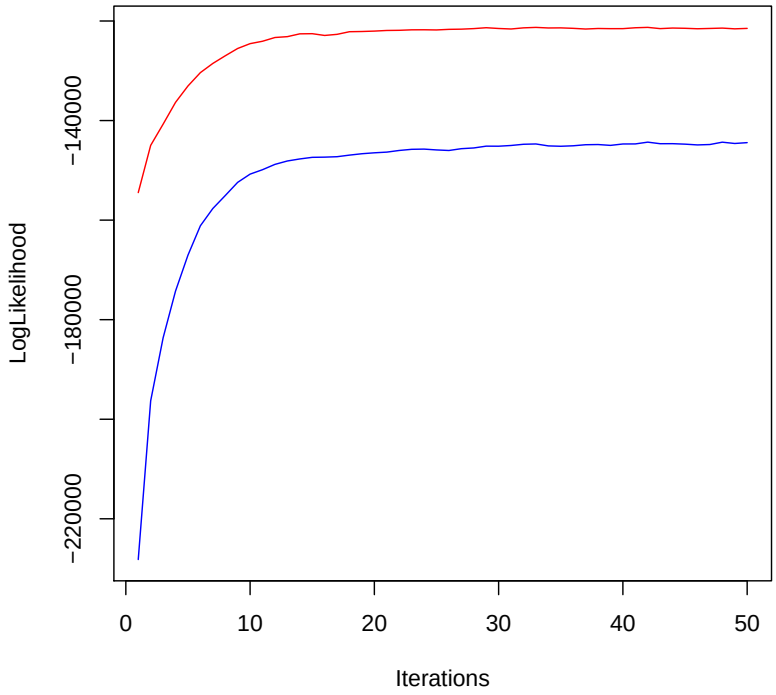


Figura 3.3: Grafico della log-verosimiglianza relativo al modello LDA. Possiamo notarne lo stabilizzarsi giá dopo 30/40 iterazioni

Il passo successivo é associare i topic individuati ai documenti (nel nostro caso i tweet); questo ci permette di capire come ogni tweet possa essere generato da uno o piú topic.

Presenteremo due tabelle: la prima identifica 7 documenti, ordinandoli, per ogni topic (tabella 3.2); nella seconda sono contenuti i testi completi dei primi tre documenti per ogni topic. (L'ultimo comando mostra le proporzioni di mistura, l'output non viene mostrato per questioni di leggibilitá)

```
top.docs=top.topic.documents(result$document_sums,
num.documents=7, alpha = 0.1)
topic.proportions=t(result$document_sums)
        /colSums(result$document_sums)
```

	1	2	3	4	5
1	non	caterina	sperimentazione	sperimentazione	stampa
2	sperimentazione	#ricerca	animale	animale	web
3	fermi	davvero	legge	nasce	insidie
4	cattaneo	#sperimentazione	decreto	ruolo	@la
5	italiano	adesso	metodi	farmaco	ministero
6	animale	#vivisezione	alternativi	lorenzin	scienza
7	elena	dimostriamolo	veronesi	ministro	sperimentazione
8	governo	#sperimentazione_animale	d	@leganerd	caso
9	video	#animale	ricercatori	basta	animali
10	difende	simonsen	prima	sí	vicolo
	6	7	8	9	10
1	#vivisezione	sperimentazione	italia	senato	italia
2	#sperimentazione_animale	non	ue	sperimentazione	non
3	non	animale	na	animale	sperimentazione
4	@orianoper	ricercatori	denunciare	convegno	ue
5	scientifica	vivisezione	sperimentazione	diretta	corte
6	medici	manifesti	vuole	fvg	giustizia
7	disastri	fare	animale	incontro	piú
8	#sperimentazione	ancora	@wireditalia	@animalisti	animale
9	#animali	animali	caterina	conoscenza	#animali
10	limav	oggi	invitata	ricercatore	denuncia

Tabella 3.1: Composizione dei 10 topic individuati dal modello

	1	2	3	4	5	6	7	8	9	10
1	1132	1775	345	1183	1473	1979	842	1856	687	1860
2	774	1802	317	1182	1150	1976	1255	114	1082	83
3	646	1335	319	569	55	1977	400	142	1125	1730
4	652	1341	318	1527	1165	1980	475	1760	263	1908
5	660	1777	389	1528	1474	668	869	1837	333	1914
6	769	27	392	1529	29	1589	304	126	686	1940
7	785	102	393	1530	414	265	521	1737	1500	1941

Tabella 3.2: Documenti con relativi topic. Ogni numero identifica una riga del dataset, quindi un tweet

L'analisi LDA sembra aver portato buoni risultati: in tabella 3.3 sono presentati i tweet. Vediamo qualche esempio: i tweet relativi al topic 7 fanno riferimento ai metodi di ricerca; il topic 4 fa riferimento ai retweet di orianoPER. Da notare che anche gli altri retweet, che non erano stati individuati dalle analisi precedenti, vengono accorpati negli stessi topic. Rimane comunque un problema fondamentale: i topic individuati da questo modello catturano la correlazione tra le parole, ma non quella tra topic. Questa limitazione deriva dal fatto di estrarre le proporzioni di topic da una singola *Dirichlet*; come conseguenza, il modello non riesce a gestire i casi in cui i topic co-occorrono frequentemente. Nei dati testuali reali invece le correlazioni tra argomenti sono comuni, e l'ignorarle porta ad una scarsa capacità predittiva, oltre alla mancata capacità di creare topic specifici, e il rischio di crearne privi di senso. Nel prossimo capitolo si discuteranno metodi piú completi, che riescono a gestire anche la correlazione tra topic.

3.4 Uteriori sviluppi

Di recente è stato proposto un nuovo topic model specifico per i tweet, il Twitter-Network (TN) (Lim, Chen, and Buntine, 2013); il TN fa uso degli hashtag, degli autori e dei followers per modellare in modo ottimale i tweet. É composto da due parti: un topic model basato su un processo Poisson-Dirichlet gerarchico (HPDP) per i testi e gli hashtag, e un modello casuale basato su un processo gaussiano (GP) per la rete di follower; i due processi vengono poi connessi attraverso le informazioni sugli autori.
Il modello HPDP si sviluppa come segue:

1. Si genera la distribuzione globale dei topic μ_0 che verrà utilizzata come a priori

2. Si genera la distribuzione dei topic v per ogni autore e una distribuzione su un a mistura di topic μ_1 per carpire eventuali scostamenti dagli argomenti discussi più di frequente dall'autore

3. Data ν e μ_1 si genera la distribuzione dei topic sui documenti e sulle parole (η, θ', θ).

Top Tweet		
Topic	Order	Tweet
1	1	Domani al Senato incontro sulla #sperimentazione animale, diretta streaming dalle 10.00 su @le_scienze programma: http://t.co/lxxmeKTidH
	2	#Lav Michela Kuan, spiega perché non parteciperá al Convegno sulla sperimentazione animale al Senato http://t.co/qOUg42Ned5
	3	La lettera con cui @AlePapale, scienziato di #SEL, prova a fermare la deriva antiscientifica della #sinistra italiana http://t.co/kmb2v9MEgy
2	1	Scopri tutti i prodotti della linea da Colpo Di Coda!!
	2	RT @NPagnoncelli: I risultati della ricerca che ho presentato oggi al Senato "Le opinioni degli italiani sulla sperimentazione animale" htt?
	3	Ogni volta che la #brambilla parla di sperimentazione animale, un neurone muore, una laurea prende fuoco. Dille di smettere anche tu.
3	1	RT @orianoPER: http://t.co/oq2MZadWjJ Il mondo #cattolico pro #vivisezione. Dobbiamo fare sentire a questo mondo la voce di chi non ha voce
	2	Paolo Bernini (M5S) ce lo dice l'europa... http://t.co/GyJVcurMrC
	3	RT @CorriereAnimali: http://t.co/8PtTnOemIQ. Sul forum si parla si sperimentazione animale, metodi alternativi e... http://t.co/M9bwtRczMo
4	1	RT @orianoPER: http://t.co/E0t71HlBDf Metodi alternativi alla #sperimentazione #animale / #garattini in #Medicina #Farmacologia #Chimica
	2	RT @orianoPER: http://t.co/fGzpz2zZ6a La #ricerca in #Medicina senza #vivisezione / #sperimentazione animale é giá il presente,va finanziat?
	3	Decreto sulla sperimentazione animale. L'analisi d'impatto del ministero della Salute. Commissione Ue Senato: ... http://t.co/TFWlljqN3K
5	1	Poveri piccolini .... Abruzzo ... Qualcuno puó aiutarli ? http://t.co/Pwr34uMi19
	2	http://t.co/BqiLKrY53S Articoli scientifici accreditati x la #ricerca senza #sperimentazione animale validata per specie umana.#Farmacologia
	3	Se stiamo con Caterina fermiamo la legge sulla sperimentazione animale http://t.co/Wd0zqPkdfO
6	1	ECCO PERCHE' SI PUO' FARE SENZA LA SPERIMENTAZIONE ANIMALE gt;http://t.co/igyGXIzyDS
	2	Le ragioni dietro la sperimentazione animale http://t.co/9RwDQ877mA via @wireditalia
	3	Elena Cattaneo: "Il Governo italiano non fermi la sperimentazione animale" (VIDEO) http://t.co/BzF33Uxbtt via @HuffPostItalia
7	1	RT @SorryNs: Chi ha detto che non esiste un metodo alternativo alla sperimentazione animale?
	2	RT @fenzi82: ?@ftinazzo: #Sperimentazioneanimale: il vicolo cieco della #scienzabiomedica #vivisezione http://t.co/rItikBO7e5? #NWO @Enrico?
	3	"Sperimentazione animale": dentro al laboratorio http://t.co/aeu31JYXK
8	1	Sperimentazione animale - Gli Eurodeputati Sonia Alfano e Andrea Zanoni scrivono al Ministro Lorenzin - http://t.co/LoRGBaFsI3
	2	http://t.co/C0zVMykOW5 USA: #ricerca tossicologica del 21'secolo senza #vivisezione /#sperimentazione #animali xché non predittive x uomo
	3	Se davvero stiamo con Caterina, dimostriamolo. Adesso http://t.co/S1Mluy0wL8
9	1	Pro o contro la sperimentazione animale? http://t.co/zCTzs5E1gz
	2	RT @kitty_chanel: Sono assolutamente convinta che se i difensori a spada tratta della sperimentazione animale devolvessero il... http://t.c?
	3	RT @orianoPER: http://t.co/HVL3T8S3QX Tumori da amianto? x la #sperimentazione su cavie e #vivisezione non risulta nocivo! Spiegatelo ai ma?
10	1	@AchilleNobiloni Convegno sulla sperimentazione animale e diritto alla conoscenza e alla salute promosso dalla commissione sanitá del Senato
	2	Se davvero stiamo con Caterina, dimostriamolo. Adesso: Il pensiero di Caterina Simonsen, studen... http://t.co/2FHc9ZqB1a via @chefuturo
	3	http://t.co/gZLeGfAubz

Tabella 3.3: La tabella presenta i primi tre tweet associati ad ognuno dei 10 topic trovati dal modello

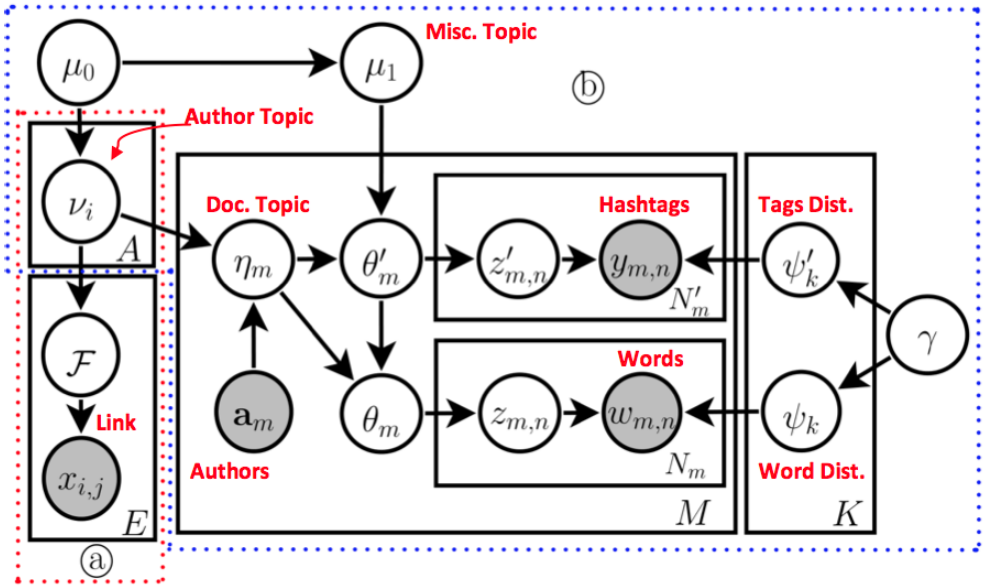


Figura 3.4: Rappresentazione grafica del modello Twitter-Network

Viene inoltre modellata l'influenza degli hashtag sulle parole; la generazione di hashtag e parole segue una LDA standard. La modellazione della rete di tweet è connessa al processo HPDP grazie alla distribuzione ν dei topic sugli autori, dove ν viene utilizzata come input per il processo gaussiano nel modello network. Il GP, denotato con F, determina le connessioni tra gli autori (x). In figura 3.4 è rappresentato il modello grafico del Twitter-Network: le regioni a e b sono rispettivamente il modello network e il topic model. Non é ancora stato presentato un applicativo per la modellazione del Twitter-Network, ci limitiamo quindi alla sola esposizione teorica.

Capitolo 4

Topic Model su corpus di articoli

Le analisi fatte finora si limitano ad una fase preliminare, l'informazione che possiamo trarne dai soli tweet é si importante ma poco variegata. Va inoltre sottolineato che i tweet raramente soddisfano le ipotesi di mistura, fondamentale per il modello LDA: in un numero cosí ristretto di termini gli argomenti trattati non possono certo esser molti. Cerchiamo di spingerci oltre: spostiamo il nostro interesse dai tweet agli articoli estratti dagli url presenti in questi ultimi. In questo modo ampliamo la quantitá di testo disponibile, con un conseguente arricchimento sia del linguaggio sia degli argomenti. Anche i modelli diventano piú complessi, allo scopo di riuscire a carpire tutte le sfumature linguistiche e grammaticali, fornendo come risultato topic piú specifici.

4.1 Creazione del corpus di articoli

La procedura per l'estrazione è semplice anche se un po' lenta. Per prima cosa estraiamo le stringhe corrispondenti agli short-url dai tweet e introduciamo la funzione `short2longURL`,che si occuperà di espanderli.

```
url=str_extract(tw$text,"http([[:graph:]]+)|www\\.([[:graph:]]+)")

short2longURL=function (url, ...)
{
  request_url = paste("http://expandurl.appspot.com/expand?url=",
                      url, "&format=json", sep = "")
  return(fromJSON(getURL(request_url, useragent = "twitteR",
                         ...))[["end_url"]])
}
```

A questo punto applichiamo la funzione sopra, `short2longURL`, a tutti gli url (escludendo quelli diversi da NA ovviamente):

```
url=unique(subset(urls,!is.na(urls)))
dec=c(1:length(url))
for(i in 1:length(url)){
  cat(" ciclo ",i)
  if(url=="NA"){
    dec[i]=0
  }
  else{
  dec[i]=decode(url[i])
  }
  save(dec,file="dec.RData")
}
```

Ora che gli url sono stati estratti ed espansi, vanno trasformati in modo che la funzione `ArticleExtractor` (presente nella libreria boilerpipeR (Annau, 2014)) li riconosca, in particolare vengono trasformati in URI (anche qui eliminiamo quelli che danno errori):

```
content=c(1:length(unique(decode)))
for(i in 1:length(unique(decode))){
    content[i]=try(getURL(unique(decode[i])))
```

```
    if(!is(content,"try-error")){
    }
    else{
        content[i]=0
    }
}
```

Siamo arrivati alla fase conclusiva, quella di estrazione del testo, la funzione utilizzata è quella accennata poco sopra:

```
documents=c(1:length(content)
for(i in 1:length(content)){
    documents[i]=ArticleExtractor(content[i])
}
```

Il vettore che ne risulta é cosí formato:

```
str(documents)
che [1:671] "Piu informazione su: malattie genetiche, medicina,
ricerca scientifica, sperimentazione animale.
Biologa, ricercatrice, ma..."
```

Il dataset é presente nella libreria `TextWiller`.

4.2 LDA

Procediamo ora all'analisi vera e propria, ripercorrendo i passaggi del capitolo precedente: prima di tutto la normalizzazione del testo e quindi la lessicalizzazione (come prima, viene scelto appositamente un vocabolario di parole che non contenga stopword italiane e termini presenti meno di 3 volte).

```
data(ArticoliSperimentazioneanimale)
documents=ArticoliSperimentazioneanimale
corpus.doc=normalizzaTesti(documents,normalizzacaratteri=TRUE,
        tolower=TRUE,perl=FALSE,fixed=FALSE)
corpus1 = lexicalize(corpus.doc$testo)
to.keep.voc = corpus1$vocab[word.counts(corpus1$documents,
        corpus1$vocab) >= 3]
to.keep.stop=subset(to.keep.voc,is.na(pmatch
        (to.keep.voc,itastopwords)))
corpus.doc= lexicalize(corpus.doc$testo,vocab=to.keep.stop)
```

É giunto il momento di calcolare il modello: si inizializzano come sempre le variabili N, K, Top e I e quindi i comandi grafici per controllare lo stabilizzarsi della funzione di verosimiglianza (figura 4.1):

```
N   = nrow(corpus.doc)
K   = 10
Top = 10
I   = 50
result = lda.collapsed.gibbs.sampler(corpus.doc, K,
        to.keep.stop, I, 0.1, 0.1, compute.log.likelihood=TRUE)
plot(c(1:I),result$log.likelihoods[1,],type="l",ylim=c(min(min
        (result$log.likelihoods[1,]),
        min(result$log.likelihoods[2,])),
        max(max(result$log.likelihoods[1,]),
        max(result$log.likelihoods[2,]))),
        col="blue",xlab="Iterations",ylab="LogLikelihood")
lines(result$log.likelihoods[2,],col="red")
```

Il modello ha raccolto dieci topic, ognuno dei quali composto da dieci termini. Il risultato é quello presentato in tabella 4.1.

	1	2	3	4	5
1	accedi	the	animali	gravidanza	non
2	google	of	sperimentazione	studi	piú
3	youtube	and	salute	mg	umani
4	video	to	sperimentazioni	durante	animali
5	account	for	ministero	dosi	vita
6	twitter	that	punti	somministrazione	specie
7	immagini	or	legge	non	esseri
8	milano	emotelove	alternativi	ratti	spesso
9	?	human	test	sicurezza	mai
10	gmail	animal	oggi	allattamento	umana
	6	7	8	9	10
1	commissione	é	sperimentazione	animali	false
2	direttiva	animale	caterina	farmaci	?false?
3	europea	cattaneo	animale	modelli	lsdexception
4	ue	ricerca	simonsen	metodi	locked
5	2010	topi	scienza	farmaco	priority
6	l'italia	sperimentazione	non	cellule	semihidden
7	corte	scientifico	caso	alternativi	unhidewhenused
8	63	animalisti	ricerca	umano	name
9	italia	governo	partito	ricerca	accent
10	senato	cani	fatto	possono	1

Tabella 4.1: Composizione dei 10 topic piú rappresentativi

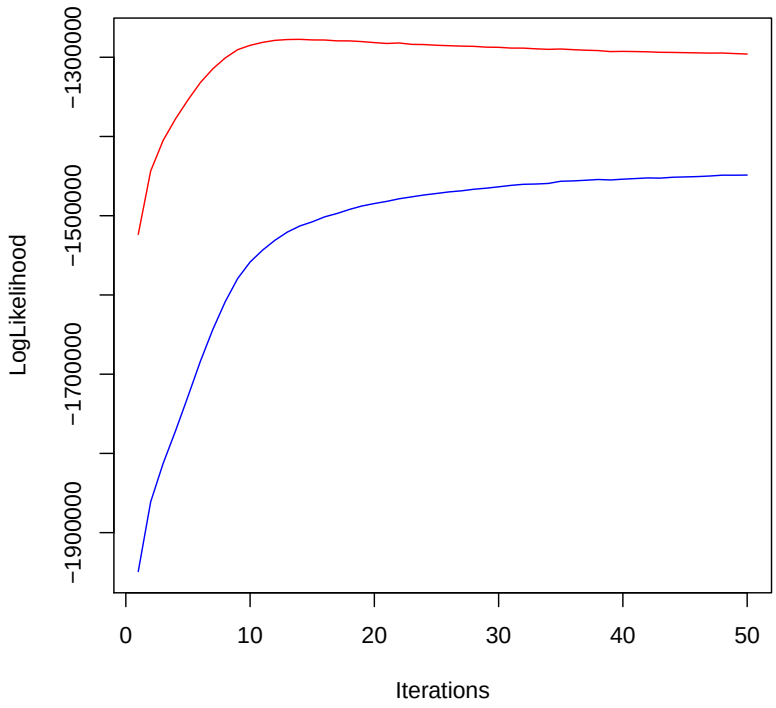


Figura 4.1: Grafico della log-verosimiglianza relativo al modello LDA. Possiamo notarne lo stabilizzarsi giá dopo le 30 iterazioni

4.3 Pachinko Allocation Model (PAM)

Il modello LDA funziona abbastanza bene, ma come giá accennato non riesce a gestire le correlazioni tra topic: puó capitare che due argomenti co-occorrano frequentemente. Per sopperire a questa mancanza è stato proposto un modello più generale, il PAM, *Pachinko allocation model* (Wei and Andrew, 2007b): il nome è preso da un gioco d'azzardo giapponese, il pachinko per l'appunto, in cui sfere di metallo cadono attraverso una complessa rete di ostacoli fino ad atterrare su pulsanti posti alla base. Questo modello fa uso di una struttura a *grafo aciclico diretto* (DAG) per rappresentare e incorporare la possibilità di topic annidati e di correlazione tra essi; un grafo aciclico diretto è un particolare tipo di grafo diretto che non ha cicli diretti, ovvero comunque scegliamo un vertice del grafo non possiamo tornare ad esso percorrendo gli archi del grafo. Un grafo diretto può dirsi aciclico (cioè è un DAG) se non presenta archi all'indietro. Nella struttura DAG, ogni nodo-foglia è associato ad una parola nel vocabolario, ed ogni nodo superiore corrisponde ad un topic, avente una distribuzione sui nodi figli. Un nodo interno che abbia tutti nodi foglia come figli corrisponde alla LDA tradizionale. Nel PAM, il concetto di topic è esteso ad essere non solo una distribuzione sulle parole, ma anche sugli altri topic; lascia però la possibilità ai nodi superiori di avere come nodi figli altri topic, rappresentanti una mistura di questi, catturando così tutte le possibili correlazioni (figura 4.2).

Nel modello PAM la distribuzione di ogni nodo interno può essere parametrizzata arbitrariamente; prendiamo in considerazione la parametrizzazione tramite un vettore della stessa dimensione del numero di figli, estratto da una *Dirichlet*. Per generare un documento si procede come segue:

1. Si estrae $\theta_{t_1}^{(d)}, \theta_{t_2}^{(d)}, ..., \theta_{t_s}^{(d)}$ da $g_1(\alpha_1), g_2(\alpha_2), ..., g_s(\alpha_s)$, dove $\theta_{t_i}^{(d)}$ è una multinomiale del topic t_i sui figli

2. Per ogni parola w nel documento

   (a) Si sceglie una partizione z_w di lunghezza $L_w :< z_{w1}, z_{w2}, ..., z_{wL_w} >$. z_{w1} è sempre il nodo radice, da z_{w2} a z_{wLw} sono i nodi dei topic T. z_{wi} è nodo figlio di $z_{w(i-1)}$ ed è estratto dalla distribuzione multinomiale $\theta_{z_w(i-1)}$

   (b) Si estrae una parola w da $\theta_{z_w L_w}$

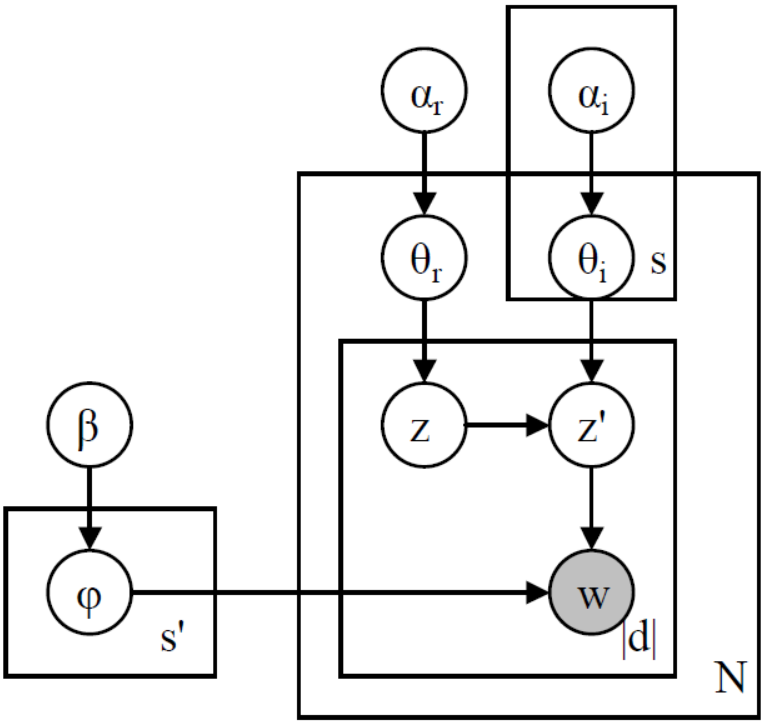


Figura 4.2: Il modello grafico relativo al PAM

Il modello generale che ne risulta è:

$$\Pr(\boldsymbol{D}|\alpha) = \prod_d \Pr(d|\alpha)$$

Confrontiamo ora graficamente i due modelli appena proposti, visualizzati in figura 4.3. Le figure rappresentano rispettivamente:

(a) Multinomiale-Dirichlet: per ogni documento, una distribuzione multinomiale sulle parola è estratta da una singola Dirichlet.

(b) LDA: si estrae una moltinomiale sui topic per ogni documento, e quindi si generano le parole dai topic.

(c) PAM a 4 livelli: la gerarchia consiste di una radice, un insieme di supertopic, un insieme di sub-topic e un vocabolario. Le radici e i supertopic sono associati a distribuzioni di Dirichlet, e da esse si estraggono le multinomiali sui nodi figli per ogni documento.

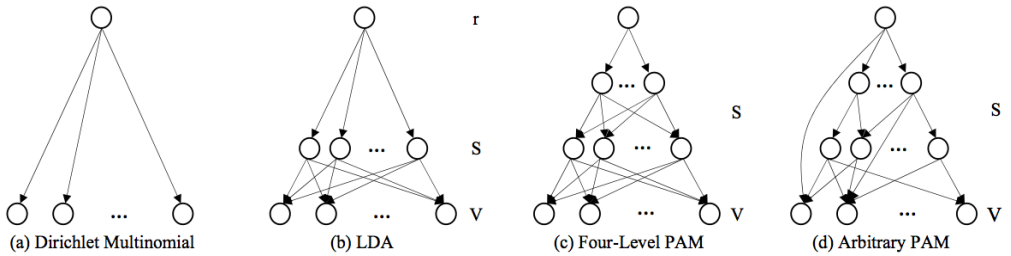


Figura 4.3: Modelli grafici per la generazione di Multinomiale-Dirichlet, LDA, PAM e PAM a 4 livelli. (a) Multinomiale-Dirichlet: per ogni documento, una distribuzione multinomiale sulle parola è estratta da una singola Dirichlet. (b) LDA: si estrae una moltinomiale sui topic per ogni documento, e quindi si generano le parole dai topic. (c) PAM a 4 livelli: la gerarchia consiste di una radice, un insieme di super-topic, un insieme di sub-topic e un vocabolario. Le radici e i super-topic sono associati a distribuzioni di Dirichlet, e da esse si estraggono le multinomiali sui nodi figli per ogni documento. (d) PAM: ha una struttura DAG arbitraria per gestire le correlazioni. Ogni nodo interno è considerato topic e associato ad una distribuzione di Dirichlet.

(d) PAM: ha una struttura DAG arbitraria per gestire le correlazioni. Ogni nodo interno è considerato topic e associato ad una distribuzione di Dirichlet.

E' inoltre possibile una generalizzazione di tipo gerachico per i modelli appena esposti, che verranno presentati più avanti: hLDA e hPAM. Ora utilizziamo il modello PAM appena proposto per identificare i topic nel dataset di documenti. Non è ancora stata sviluppata una libreria in R che implementi questo modello; un metodo alternativo che si propone é di utilizzare R come wrapper di una libreria Java denominata *mallet*, scritta dagli stessi autori di questi topic model. I comandi che seguono mostrano la procedura passo passo.

4.3.1 Implementazione del modello PAM

Iniziamo con il definire la directory in cui è situato il pacchetto mallet scaricabile dal sito (McCallum, 2002):

```
dir = "~/mallet-2.0.7"
setwd(dir)
```

I file di testo usati finora erano un vettore di tipo `chr` in R, mallet necessità di tanti file in formato txt quanti sono i documenti; con R la procedura per creare questi file è molto semplice:

```
for(i in 1:length(documents)){
  write.table(documents[i],
  file=paste("articolo",i,".txt",sep=""))
}
```

Una volta creati i documenti necessari, definiamo il percorso della cartella in cui sono situati, in modo che R (e quindi il terminale) possa importarli:

```
importdir = "/Users/Federico/Desktop/Tesi/TestiMallet"
```

Vanno definite alcune variabili che serviranno nei successivi processi, in particolare il nome del file che fungerà da training, i parametri del modello (numero di topic e intervallo di ottimizzazione) e i nomi dei file di outoput.

```
output = "tutorial.mallet"
ntopics = 20
optint =  20
outputstate =  "topic-state.gz"
```

Le variabili sono state create, vanno ora combinate appositamente in una stringa che verrà inviata al terminale

```
cd = "cd ~/mallet-2.0.7" # location of the bin directory
import = paste("bin/mallet import-dir --input", importdir,
        "--output", output, "--keep-sequence
        --remove-stopwords", sep = " ")
train  = paste("bin/mallet train-topics  --input", output,
        "--num-topics", ntopic,"--use-pam", "true",
        "--optimize-interval",  optint, "--output-state",
        outputstate, sep = " ")
MALLET_HOME = "~/mallet-2.0.7"
```

I passaggi preliminari sono stati eseguiti, non resta che inviare i comandi al terminale in modo che proceda al calcolo del modello:

```
#Per sistemi Windows
Sys.setenv("MALLET_HOME" = MALLET_HOME)
```

```
Sys.setenv(PATH = "c:/Program Files (x86)/Java/jre7/bin")

shell(shQuote(paste(cd, import, train, sep = " && ")),
         invisible = FALSE)

#Per sistemi Unix
system(paste(cd, import, train, sep = " ; "))
```

In tabella 4.2 sono presentati i risultati del modello PAM (da notare la mancanza delle lettere accentate, questo é un problema di mallet nel gestire la lingua italiana). Vediamone qualche esempio: i topic 1, 3 e 14 raccolgono tutti i termini in lingua inglese, ed il topic 1 in particolare la maggior parte delle stopword; i topic 4 e 20 si riferiscono alla parte normativo-legislativa della questione sperimentazione animale; il topic 2 sembra essere molto legato ad articoli di cronaca, simili a quello esposto nell'introduzione. Possiamo ritenerci soddisfatti, i topic sembrano essere specifici ed esplicativi. Proseguiamo ora con altri due modelli, limitandoci alla sola presentazione teorica.

4.4 Hierarchical LDA (hLDA)

Il modello LDA *gerarchico* (Blei, Mimno, Griffiths, T.L., Jordan, M.I., Tenenbaum, and B., 2004) rappresenta la distribuzione dei topic presenti nei documenti organizzando gli stessi in un albero: ogni documento è generato dai topic presenti in una singola partizione dell'albero. Nella fase di apprendimento, il campionamento si alterna tra la scelta di una nuova partizione dell'albero per ogni documento e l'assegnazione di ogni parola in ogni documento ad un topic appartenente alla partizione scelta. La struttura dell'albero è dedotta dagli stessi topic attraverso l'utilizzo di un modello *Chinese Restaurant Process*.

4.4.1 Il Chinese Restaurant Process

In teoria delle probabilità, il Chinese Restaurant Process è un processo stocastico discreto, il cui valore in qualsiasi momento n è una partizione $B(n)$ dell'insieme $\{1, 2, 3, ..., n\}$ la cui distribuzione di probabilità è determinata come segue. Al tempo $n = 1$, la partizione banale $\{\{1\}\}$ è ottenuta con probabilità 1, al tempo $n + 1$ l'elemento $n + 1$ può essere:

1. inserito in uno dei blocchi della partizione $B(n)$, dove ogni blocco viene scelto con probabilità $\left(\frac{|b|}{n+1}\right)$, dove $|b|$ è la dimensione del blocco;

2. oppure aggiunto alla partizione $B(n)$ come un nuovo blocco singolo, con probabilità $\frac{1}{n+1}$. La partizione casuale così generata è scambiabile, cioè riclassificando $\{1, ..., n\}$, non viene modifica la distribuzione di probabilità della partizione, ed è coerente nel senso che la legge della partizione di $n-1$ ottenuta rimuovendo l'elemento n dalla partizione casuale al tempo n è la stessa legge della partizione casuale al tempo $n-1$.

Rendiamo più eplicita la metafora. Si immagini un ristorante vuoto con un numero potenzialmente infinito di tavoli. In questo processo si suppone che vi sia un flusso infinito di clienti. Quando un cliente arriva può essere fatto accomodare ad un tavolo già esistente o ad un tavolo vuoto. Qualora il cliente sia fatto aggregare ad un tavolo già presente questo riceverà il medesimo piatto che hanno gli altri clienti a quel tavolo, se invece viene fatto sedere ad un tavolo libero gli verrà portato un piatto a scelta dallo chef tra quelli non ancora proposti. É possibile estendere il processo considerando un franchising di ristoranti cinesi. In questa nuova configurazione si può immaginare di muoversi su più livelli in cui il massimo focus lo si trova considerando il singolo documento (ristorante nella metafora) e, viceversa, la vista generale verrà fornita osservando l'intero corpus di documenti (una catena di ristoranti). Si suppone esistano J ristoranti in franchising. I clienti del $j-$esimo ristorante vengono fatti accomodare con la stessa dinamica del Chinese Restaurant Process e ciò accade in maniera indipendente per ogni ristorante. Quando un cliente entra in un ristorante infatti può essere fatto accomodare ad un tavolo già esistente o ad un tavolo vuoto. Qualora il cliente sia fatto aggregare ad un tavolo già presente questo riceverà il medesimo piatto che hanno gli altri clienti a quel tavolo, se invece viene fatto sedere ad un tavolo libero gli verrà portato un piatto a scelta dallo chef. A differenza del CRP qui il menù è condiviso fra tutti i ristoranti della catena. In questo caso quindi tavoli diversi in ristoranti diversi potranno condividere lo stesso piatto, ma allo stesso modo anche tavoli all'interno dello stesso ristorante potranno essere caratterizzati dalla stessa pietanza. A differenza del CRP qui i tavoli non rappresentano più cluster differenti ma la loro identità viene fornita dal piatto mangiato.

4.5 Hierarchical PAM (hPAM)

Nel modello PAM *gerarchico* (David, Wei, and Andrew, 2007) ad ogni nodo, non solo quelli terminali, è associata una distribuzione sul vocabolario; questa generalizzazione porta al raggiungimento di un'estrema flessibilità per il topic modeling. Saranno presentate due varianti, ma verrà analizzata sui dati solo la seconda.

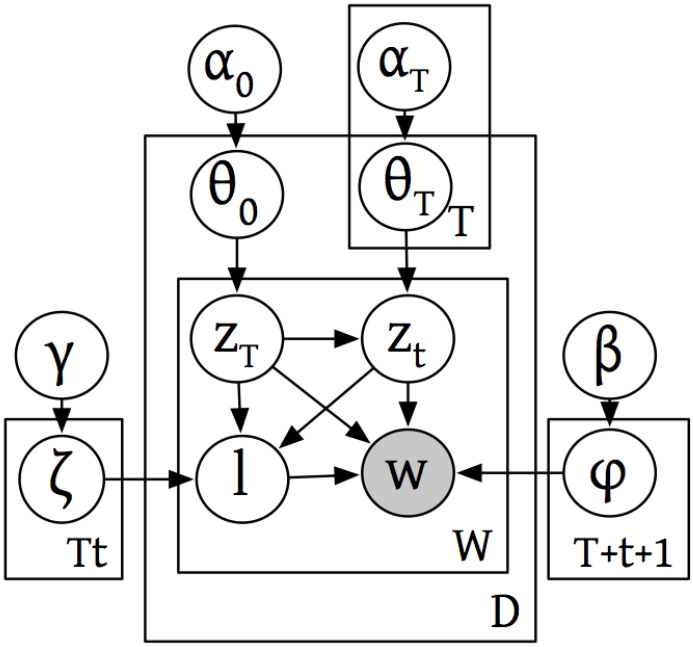


Figura 4.4: Modello grafico per un generico hPAM

Nella prima variante, denominata hPAM1, ad ogni partizione dell'albero è associata una distribuzione sui livelli della partizione stessa. Il processo di generazione per un documento è il seguente:

1. Per ogni documento d, si estrae una distribuzione θ_0 sui super-topic e una distribuzione θ_T sui sub-topic per ogni super-topic

2. Per ogni parola w,

   (a) Si estrae un super-topic z_T da θ_0

   (b) Si estrae un sub-topic z_t da θ_{z_T}

(c) Si estrae un livello l da $\zeta_{z_T z_t}$

(d) Si estrae una parola da ϕ_0 se $l = 1$, da ϕ_{z_T} se $l = 2$, da ϕ_{z_t} se $l = 3$

La seconda variante, hPAM2, è simile alla prima, ma non comprende le distribuzioni sui livelli; la distribuzione di *Dirichlet* contiene però, per ogni nodo interno, una dimensione extra. Questa dimensione aggiuntiva corrisponde alla possibilità che una parola sia estratta direttamente dal nodo interno, senza mai raggiungere i nodi-foglia. Il processo generativo è mostrato in figura 4.4.

1. Per ogni documento d, si estrae una distribuzione θ_0 sui super-topic e una distribuzione θ_T sui sub-topic per ogni super-topic

2. Per ogni parola w,

   (a) Si estrae un super-topic z_T da θ_0. Se $z_T = 0$, si estrae una parola da ϕ_0

   (b) Altrimenti, si estrae un sub-topic z_t da θ_{z_T}. Se $z_T = 0$, si estrae una parola da ϕ_{z_T}

   (c) Altrimenti, si estrae una parola da ϕ_{z_t}

In figura 4.5 vengono presentati graficamente quattro modelli, nello specifico:

(a) Modello LDA gerarchico

(b) Modello PAM a 4 livelli

(c) Modello PAM gerarchico 1, ad ogni partizione dell'albero é associata una distribuzione sui livelli della partizione stessa

(d) Modello PAM gerarchico 2, non comprende le distribuzioni sui livelli

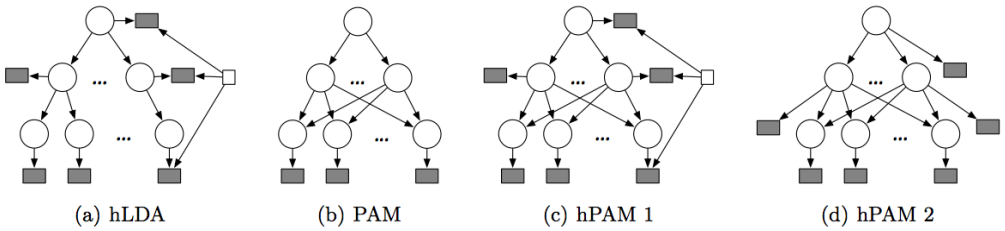


Figura 4.5: Modelli grafici per la generazione di hLDA, PAM, e hPAM 1 e 2. hLDA e hPAM includono ditribuzioni multinomiali sulle parole (rappresentate dai rettangoli grigi) ad ogni nodo, con distribuzioni separate sui livelli per ogni partizione (rappresentate dai rettangoli bianchi). hLDA ha una struttura ad albero: un singolo topic per ogni livello è connesso ad uno del livello più basso. PAM e hPAM sono caratterizzati da una struttura DAG, quindi ogni nodo di un dato livello ha una distribuzione sui nodi del livello più basso.

	Topic 1	Proportion	Topic 2	Proportion	Topic 3	Proportion	Topic 4	Proportion	Topic 5	Proportion
1	the	0,05416	universit	0,01319	dr	0,02553	direttiva	0,05355	europea	0,04465
2	of	0,04409	milano	0,01266	animal	0,02398	italia	0,04897	animali	0,0397
3	and	0,04272	animale	0,01159	http	0,01799	commissione	0,03424	pi	0,02733
4	to	0,03527	proprio	0,01079	you	0,01412	ue	0,0272	legge	0,02391
5	for	0,02771	simonsen	0,00999	www	0,01373	non	0,02486	corte	0,01567
6	that	0,01365	immagini	0,00973	on	0,01219	paese	0,01483	fini	0,01567
7	or	0,0106	dati	0,00933	drug	0,01122	gennaio	0,01429	nazionale	0,01472
8	human	0,01008	dopo	0,00906	studi	0,01025	scientifici	0,01205	ancora	0,01414
9	is	0,00777	alcuni	0,0088	can	0,0089	recepimento	0,01195	normativa	0,0139
10	it	0,00766	associazione	0,0084	with	0,00851	protezione	0,01195	stati	0,01296
11	have	0,00766	minacce	0,008	nih	0,00735	decreto	0,01163	stato	0,0119
12	page	0,00672	stati	0,00746	your	0,00735	utilizzati	0,0112	parere	0,01131
13	we	0,00619	italia	0,00733	health	0,00677	articolo	0,01099	esperimenti	0,01084
14	by	0,00609	cento	0,00706	it	0,00677	secondo	0,01035	infatti	0,01013
15	models	0,00609	altro	0,0068	journal	0,00677	giorno	0,01013	testo	0,00978
16	are	0,00609	facebook	0,00653	vol	0,00677	parlamento	0,01003	europeo	0,00942
17	full	0,00577	pubblico	0,00586	new	0,006	euro	0,00971	gi	0,00931
18	research	0,00577	padovan	0,00586	drugs	0,0058	essere	0,00939	norme	0,00883
19	testing	0,00577	sito	0,00573	med	0,0058	legislativo	0,00928	diritto	0,00872
20	as	0,00567	giorno	0,00573	are	0,00561	attuazione	0,00928	ricercatori	0,00848
	Topic 6	Proportion	Topic 7	Proportion	Topic 8	Proportion	Topic 9	Proportion	Topic 10	Proportion
1	animali	0,03797	pi	0,01804	non	0,1118	non	0,04183	animale	0,09378
2	sperimentazione	0,02959	ricerca	0,01775	perch	0,03261	umani	0,02423	sperimentazione	0,09327
3	ricerca	0,02398	persone	0,01679	cosa	0,02085	specie	0,02046	parte	0,01873
4	animale	0,02285	non	0,01497	animalisti	0,02085	test	0,01727	senza	0,01657
5	salute	0,02154	ricercatori	0,01439	fatto	0,02031	animali	0,01473	vivisezione	0,01621
6	senato	0,01528	animali	0,01276	caso	0,01856	metodi	0,01301	anni	0,01585
7	metodi	0,01236	solo	0,01046	caterina	0,01726	perch	0,01187	scientifica	0,01412
8	rispetto	0,00951	ora	0,00854	fare	0,01382	sperimentazione	0,01015	gennaio	0,0134
9	studi	0,00951	sempre	0,00796	essere	0,01229	milioni	0,00794	due	0,01167
10	alternative	0,00943	malati	0,0071	cos	0,01138	viene	0,00761	oggi	0,01124
11	de	0,00935	esempio	0,00691	invece	0,01046	mai	0,00737	solo	0,01044
12	incontro	0,00862	ancora	0,00653	stata	0,01008	dopo	0,00704	alternativi	0,00994
13	alternativi	0,00854	cos	0,00653	altri	0,00916	vita	0,00663	dibattito	0,00987
14	cos	0,00846	anno	0,00643	proprio	0,00871	etica	0,00647	metodi	0,009
15	diritto	0,00821	sotto	0,00624	dire	0,00863	campo	0,00614	sempre	0,00857
16	sanit	0,00789	vita	0,00605	dare	0,00787	oltre	0,00598	diritti	0,00821
17	test	0,0074	tempo	0,00576	malattie	0,00779	sviluppo	0,00581	grazie	0,00814
18	cavie	0,0074	altre	0,00576	parlare	0,00779	stesso	0,00573	scientifici	0,00814
19	commissione	0,00724	subito	0,00547	video	0,00733	laboratorio	0,00573	meno	0,00792
20	legge	0,00667	cura	0,00509	mai	0,0071	risposta	0,00565	stato	0,00771
	Topic 11	Proportion	Topic 12	Proportion	Topic 13	Proportion	Topic 14	Proportion	Topic 15	Proportion
1	non	0,02954	ricerca	0,04943	animali	0,05792	false	0,11079	stato	0,02857
2	animale	0,02818	quindi	0,02221	non	0,05608	name	0,0516	effetti	0,02579
3	sperimentazione	0,02799	scientifico	0,01394	uomo	0,02496	lsdexception	0,03609	altri	0,0173
4	ricerca	0,01927	mondo	0,01176	farmaci	0,02198	locked	0,03609	esperimenti	0,01698
5	perch	0,01336	lavoro	0,01013	farmaco	0,01404	priority	0,03609	dati	0,01681
6	animali	0,01065	medicina	0,00991	modelli	0,01354	semihidden	0,03609	stati	0,01632
7	convegno	0,01027	punto	0,00947	pu	0,01326	unhidewhenused	0,03609	durante	0,01632
8	cattaneo	0,01027	anni	0,00871	esseri	0,01184	accent	0,02659	stata	0,01159
9	malattia	0,00959	va	0,00827	modello	0,01163	twitter	0,02248	ancora	0,01045
10	scientifico	0,00939	problema	0,00817	umana	0,0112	medium	0,01646	tali	0,01012
11	ricercatori	0,00939	biomedica	0,00773	molto	0,01078	pi	0,01488	cane	0,00849
12	vita	0,00862	deve	0,0074	risultati	0,00936	cookie	0,01235	sperimentale	0,008
13	laboratorio	0,00852	paesi	0,00697	topi	0,00893	list	0,01203	due	0,00784
14	senatrice	0,00852	possa	0,00675	umano	0,00815	grid	0,0114	prodotto	0,00767
15	direttiva	0,00765	scientifica	0,00653	ricerca	0,00794	shading	0,00886	sostanze	0,00718
16	metodo	0,00726	ridurre	0,00642	malattie	0,00759	offrirti	0,00823	mercato	0,00718
17	italiana	0,00707	nessuno	0,00642	base	0,00681	personalizzare	0,00823	pp	0,00686
18	governo	0,00707	importante	0,00642	umane	0,00659	usando	0,00665	fino	0,00669
19	elena	0,00697	gi	0,00631	cancro	0,00638	web	0,00665	poich	0,00653
20	europea	0,00668	meglio	0,00599	altro	0,00624	light	0,00665	test	0,00653
	Topic 16	Proportion	Topic 17	Proportion	Topic 18	Proportion	Topic 19	Proportion	Topic 20	Proportion
1	non	0,0402	non	0,03731	pi	0,05583	animali	0,05001	animali	0,02155
2	pu	0,01269	gravidanza	0,02737	essere	0,02953	test	0,02882	vivisezione	0,01868
3	scienza	0,01135	studi	0,02229	solo	0,01762	sperimentazione	0,02829	salute	0,01832
4	qui	0,00921	accedi	0,01455	modo	0,01302	oggi	0,02211	lav	0,01676
5	sel	0,00801	google	0,01086	vengono	0,01132	sperimentazioni	0,01461	ministero	0,01209
6	credo	0,00761	uomo	0,01074	vita	0,01089	essere	0,01395	governo	0,01138
7	politica	0,00708	mg	0,01051	spesso	0,01055	prodotti	0,0129	schema	0,0097
8	animalista	0,00694	youtube	0,01005	gi	0,01013	italiani	0,0129	lorenzin	0,00934
9	forza	0,00694	ratti	0,01005	ancora	0,00902	testare	0,01237	associazioni	0,00838
10	ragione	0,00641	video	0,00982	prima	0,00894	punti	0,01132	art	0,00754
11	partito	0,00628	animali	0,00924	molti	0,00842	scientifica	0,01079	presidente	0,00718
12	interno	0,00614	sicurezza	0,00924	realt	0,00834	medici	0,01079	affari	0,00707
13	temi	0,00614	sviluppo	0,00866	questione	0,00825	scopi	0,01079	ministro	0,00695
14	fine	0,00574	somministrazione	0,0082	senso	0,00817	necessaria	0,01013	legislativo	0,00671
15	essere	0,00494	essere	0,00809	fare	0,00808	accettabile	0,01013	delegazione	0,00671
16	parte	0,00494	animale	0,00774	capire	0,00774	ipsos	0,01	senza	0,00659
17	morte	0,00481	cani	0,00774	tratta	0,00766	cosmetici	0,00974	solo	0,00659
18	etica	0,00481	trattamento	0,00762	dunque	0,00715	opinione	0,00947	alternativi	0,00635
19	progresso	0,00467	dose	0,00762	molte	0,00715	molto	0,00855	tavolo	0,00611
20	gruppo	0,00467	allattamento	0,00762	esempio	0,00706	circa	0,0079	decreto	0,00587

Tabella 4.2: Topic relativi al modello PAM

1	Super-topic,0[18957]	Proportion	Topic Composition
2	5	0.0212	animali,sperimentazione,ricerca,animale,salute,senato,metodi,rispetto,studi,alternative,de,incontro,alternativi,cos,diritto,sanit,test,cavie,commissione,legge,
3	19	0.01973	animali,vivisezione,salute,lav,ministero,governo,schema,lorenzin,associazioni,art,presidente,affari,ministro,legislativo,delegazione,senza,solo,alternativi,tavolo,decreto,
4	3	0.01701	direttiva,italia,commissione,ue,non,paese,gennaio,scientifici,recepimento,protezione,decreto,utilizzati,articolo,secondo,giorno,parlamento,euro,essere,legislativo,attuazione,
5	11	0.01599	ricerca,quindi,scientifico,mondo,lavoro,medicina,punto,anni,va,problema,biomedica,deve,paesi,possa,scientifica,ridurre,nessuno,importante,gi,meglio,
6	6	0.01579	pi,ricerca,persone,non,ricercatori,animali,solo,ora,sempre,malati,esempio,ancora,cos,anno,sotto,vita,tempo,altre,subito,cura,
7	7	0.01537	non,perch,cosa,animalisti,fatto,caso,caterina,fare,essere,cos,invece,stata,altri,proprio,dire,dare,malattie,parlare,video,mai,
8	13	0.01392	false,name,lsdexception,locked,priority,semihidden,unhidewhenused,accent,twitter,medium,pi,cookie,list,grid,shading,offrirti,personalizzare,usando,web,light,
9	18	0.01354	animali,test,sperimentazione,oggi,sperimentazioni,essere,prodotti,italiani,testare,punti,scientifica,medici,scopi,necessaria,accettabile,ipsos,cosmetici,opinione,molto,circa,
10	17	0.01209	pi,essere,solo,modo,vengono,vita,spesso,gi,ancora,prima,molti,realt,questione,senso,fare,capire,tratta,dunque,molte,esempio,
11	0	0.01168	the,of,and,to,for,that,or,human,is,it,have,page,we,by,models,are,full,research,testing,as,
12	Super-topic,1[18249]	Proportion	Topic Composition
13	9	0.15737	animale,sperimentazione,parte,senza,vivisezione,anni,scientifica,gennaio,due,oggi,solo,alternativi,dibattito,metodi,sempre,diritti,grazie,scientifici,meno,stato,
14	7	0.11526	non,perch,cosa,animalisti,fatto,caso,caterina,fare,essere,cos,invece,stata,altri,proprio,dire,dare,malattie,parlare,video,mai,
15	18	0.09969	animali,test,sperimentazione,oggi,sperimentazioni,essere,prodotti,italiani,testare,punti,scientifica,medici,scopi,necessaria,accettabile,ipsos,cosmetici,opinione,molto,circa,
16	4	0.09737	europea,animali,pi,legge,corte,fini,nazionale,ancora,normativa,stati,stato,parere,esperimenti,infatti,testo,europeo,gi,norme,diritto,ricercatori,
17	1	0.09059	universit,milano,animale,proprio,simonsen,immagini,dati,dopo,alcuni,associazione,minacce,stati,italia,cento,altro,facebook,pubblico,padovan,sito,giorno,
18	11	0.09038	ricerca,quindi,scientifico,mondo,lavoro,medicina,punto,anni,va,problema,biomedica,deve,paesi,possa,scientifica,ridurre,nessuno,importante,gi,meglio,
19	5	0.08814	animali,sperimentazione,ricerca,animale,salute,senato,metodi,rispetto,studi,alternative,de,incontro,alternativi,cos,diritto,sanit,test,cavie,commissione,legge,
20	8	0.07806	non,umani,specie,test,animali,metodi,perch,sperimentazione,milioni,viene,mai,dopo,vita,etica,campo,oltre,sviluppo,stesso,laboratorio,risposta,
21	19	0.07273	animali,vivisezione,salute,lav,ministero,governo,schema,lorenzin,associazioni,art,presidente,affari,ministro,legislativo,delegazione,senza,solo,alternativi,tavolo,decreto,
22	10	0.06681	non,animale,sperimentazione,ricerca,perch,animali,convegno,cattaneo,malattia,scientifico,ricercatori,vita,laboratorio,senatrice,direttiva,metodo,italiana,governo,elena,europea,
23	Super-topic,2[18026]	Proportion	Topic Composition
24	7	0.20102	non,perch,cosa,animalisti,fatto,caso,caterina,fare,essere,cos,invece,stata,altri,proprio,dire,dare,malattie,parlare,video,mai,
25	9	0.18047	animale,sperimentazione,parte,senza,vivisezione,anni,scientifica,gennaio,due,oggi,solo,alternativi,dibattito,metodi,sempre,diritti,grazie,scientifici,meno,stato,
26	4	0.16501	europea,animali,pi,legge,corte,fini,nazionale,ancora,normativa,stati,stato,parere,esperimenti,infatti,testo,europeo,gi,norme,diritto,ricercatori,
27	5	0.12664	animali,sperimentazione,ricerca,animale,salute,senato,metodi,rispetto,studi,alternative,de,incontro,alternativi,cos,diritto,sanit,test,cavie,commissione,legge,
28	18	0.1236	animali,test,sperimentazione,oggi,sperimentazioni,essere,prodotti,italiani,testare,punti,scientifica,medici,scopi,necessaria,accettabile,ipsos,cosmetici,opinione,molto,circa,
29	10	0.12077	non,animale,sperimentazione,ricerca,perch,animali,convegno,cattaneo,malattia,scientifico,ricercatori,vita,laboratorio,senatrice,direttiva,metodo,italiana,governo,elena,europea,
30	17	0.12031	pi,essere,solo,modo,vengono,vita,spesso,gi,ancora,prima,molti,realt,questione,senso,fare,capire,tratta,dunque,molte,esempio,
31	6	0.08453	pi,ricerca,persone,non,ricercatori,animali,solo,ora,sempre,malati,esempio,ancora,cos,anno,sotto,vita,tempo,altre,subito,cura,
32	15	0.0736	non,pu,scienza,qui,sel,credo,politica,animalista,forza,ragione,partito,interno,temi,fine,essere,parte,morte,etica,progresso,gruppo,
33	11	0.05619	ricerca,quindi,scientifico,mondo,lavoro,medicina,punto,anni,va,problema,biomedica,deve,paesi,possa,scientifica,ridurre,nessuno,importante,gi,meglio,
34	Super-topic,3[18795]	Proportion	Topic Composition
35	4	0.0232	europea,animali,pi,legge,corte,fini,nazionale,ancora,normativa,stati,stato,parere,esperimenti,infatti,testo,europeo,gi,norme,diritto,ricercatori,
36	7	0.02085	non,perch,cosa,animalisti,fatto,caso,caterina,fare,essere,cos,invece,stata,altri,proprio,dire,dare,malattie,parlare,video,mai,
37	0	0.02072	the,of,and,to,for,that,or,human,is,it,have,page,we,by,models,are,full,research,testing,as,
38	9	0.01961	animale,sperimentazione,parte,senza,vivisezione,anni,scientifica,gennaio,due,oggi,solo,alternativi,dibattito,metodi,sempre,diritti,grazie,scientifici,meno,stato,
39	19	0.01663	animali,vivisezione,salute,lav,ministero,governo,schema,lorenzin,associazioni,art,presidente,affari,ministro,legislativo,delegazione,senza,solo,alternativi,tavolo,decreto,
40	5	0.01597	animali,sperimentazione,ricerca,animale,salute,senato,metodi,rispetto,studi,alternative,de,incontro,alternativi,cos,diritto,sanit,test,cavie,commissione,legge,
41	17	0.01546	pi,essere,solo,modo,vengono,vita,spesso,gi,ancora,prima,molti,realt,questione,senso,fare,capire,tratta,dunque,molte,esempio,
42	1	0.01409	universit,milano,animale,proprio,simonsen,immagini,dati,dopo,alcuni,associazione,minacce,stati,italia,cento,altro,facebook,pubblico,padovan,sito,giorno,
43	12	0.01344	animali,non,uomo,farmaci,farmaco,modelli,pu,esseri,modello,umana,molto,risultati,topi,umano,ricerca,malattie,base,umane,cancro,altro,
44	18	0.0134	animali,test,sperimentazione,oggi,sperimentazioni,essere,prodotti,italiani,testare,punti,scientifica,medici,scopi,necessaria,accettabile,ipsos,cosmetici,opinione,molto,circa,
45	Super-topic,4[19037]	Proportion	Topic Composition
46	19	0.00433	animali,vivisezione,salute,lav,ministero,governo,schema,lorenzin,associazioni,art,presidente,affari,ministro,legislativo,delegazione,senza,solo,alternativi,tavolo,decreto,
47	0	0.00411	the,of,and,to,for,that,or,human,is,it,have,page,we,by,models,are,full,research,testing,as,
48	6	0.00332	pi,ricerca,persone,non,ricercatori,animali,solo,ora,sempre,malati,esempio,ancora,cos,anno,sotto,vita,tempo,altre,subito,cura,
49	5	0.00272	animali,sperimentazione,ricerca,animale,salute,senato,metodi,rispetto,studi,alternative,de,incontro,alternativi,cos,diritto,sanit,test,cavie,commissione,legge,
50	3	0.00251	direttiva,italia,commissione,ue,non,paese,gennaio,scientifici,recepimento,protezione,decreto,utilizzati,articolo,secondo,giorno,parlamento,euro,essere,legislativo,attuazione,
51	4	0.00225	europea,animali,pi,legge,corte,fini,nazionale,ancora,normativa,stati,stato,parere,esperimenti,infatti,testo,europeo,gi,norme,diritto,ricercatori,
52	17	0.00212	pi,essere,solo,modo,vengono,vita,spesso,gi,ancora,prima,molti,realt,questione,senso,fare,capire,tratta,dunque,molte,esempio,
53	18	0.00211	animali,test,sperimentazione,oggi,sperimentazioni,essere,prodotti,italiani,testare,punti,scientifica,medici,scopi,necessaria,accettabile,ipsos,cosmetici,opinione,molto,circa,
54	12	0.00172	animali,non,uomo,farmaci,farmaco,modelli,pu,esseri,modello,umana,molto,risultati,topi,umano,ricerca,malattie,base,umane,cancro,altro,
55	9	0.00172	animale,sperimentazione,parte,senza,vivisezione,anni,scientifica,gennaio,due,oggi,solo,alternativi,dibattito,metodi,sempre,diritti,grazie,scientifici,meno,stato,
56	Super-topic,5[19877]	Proportion	Topic Composition
57	3	0.01709	direttiva,italia,commissione,ue,non,paese,gennaio,scientifici,recepimento,protezione,decreto,utilizzati,articolo,secondo,giorno,parlamento,euro,essere,legislativo,attuazione,
58	12	0.01408	animali,non,uomo,farmaci,farmaco,modelli,pu,esseri,modello,umana,molto,risultati,topi,umano,ricerca,malattie,base,umane,cancro,altro,
59	5	0.01162	animali,sperimentazione,ricerca,animale,salute,senato,metodi,rispetto,studi,alternative,de,incontro,alternativi,cos,diritto,sanit,test,cavie,commissione,legge,
60	0	0.01114	the,of,and,to,for,that,or,human,is,it,have,page,we,by,models,are,full,research,testing,as,
61	7	0.00885	non,perch,cosa,animalisti,fatto,caso,caterina,fare,essere,cos,invece,stata,altri,proprio,dire,dare,malattie,parlare,video,mai,
62	9	0.00825	animale,sperimentazione,parte,senza,vivisezione,anni,scientifica,gennaio,due,oggi,solo,alternativi,dibattito,metodi,sempre,diritti,grazie,scientifici,meno,stato,
63	8	0.00804	non,umani,specie,test,animali,metodi,perch,sperimentazione,milioni,viene,mai,dopo,vita,etica,campo,oltre,sviluppo,stesso,laboratorio,risposta,
64	4	0.00756	europea,animali,pi,legge,corte,fini,nazionale,ancora,normativa,stati,stato,parere,esperimenti,infatti,testo,europeo,gi,norme,diritto,ricercatori,
65	1	0.00718	universit,milano,animale,proprio,simonsen,immagini,dati,dopo,alcuni,associazione,minacce,stati,italia,cento,altro,facebook,pubblico,padovan,sito,giorno,
66	18	0.00697	animali,test,sperimentazione,oggi,sperimentazioni,essere,prodotti,italiani,testare,punti,scientifica,medici,scopi,necessaria,accettabile,ipsos,cosmetici,opinione,molto,circa,
67	Super-topic,6[18248]	Proportion	Topic Composition
68	5	0.00898	animali,sperimentazione,ricerca,animale,salute,senato,metodi,rispetto,studi,alternative,de,incontro,alternativi,cos,diritto,sanit,test,cavie,commissione,legge,
69	3	0.00896	direttiva,italia,commissione,ue,non,paese,gennaio,scientifici,recepimento,protezione,decreto,utilizzati,articolo,secondo,giorno,parlamento,euro,essere,legislativo,attuazione,
70	0	0.00887	the,of,and,to,for,that,or,human,is,it,have,page,we,by,models,are,full,research,testing,as,
71	9	0.00746	animale,sperimentazione,parte,senza,vivisezione,anni,scientifica,gennaio,due,oggi,solo,alternativi,dibattito,metodi,sempre,diritti,grazie,scientifici,meno,stato,
72	1	0.00641	universit,milano,animale,proprio,simonsen,immagini,dati,dopo,alcuni,associazione,minacce,stati,italia,cento,altro,facebook,pubblico,padovan,sito,giorno,
73	18	0.00509	animali,test,sperimentazione,oggi,sperimentazioni,essere,prodotti,italiani,testare,punti,scientifica,medici,scopi,necessaria,accettabile,ipsos,cosmetici,opinione,molto,circa,
74	12	0.00451	animali,non,uomo,farmaci,farmaco,modelli,pu,esseri,modello,umana,molto,risultati,topi,umano,ricerca,malattie,base,umane,cancro,altro,
75	19	0.00435	animali,vivisezione,salute,lav,ministero,governo,schema,lorenzin,associazioni,art,presidente,affari,ministro,legislativo,delegazione,senza,solo,alternativi,tavolo,decreto,
76	2	0.00434	dr,animal,http,you,www,on,drug,studi,can,with,nih,your,health,it,journal,vol,new,drugs,med,are,
77	17	0.0042	pi,essere,solo,modo,vengono,vita,spesso,gi,ancora,prima,molti,realt,questione,senso,fare,capire,tratta,dunque,molte,esempio,
78	Super-topic,7[18584]	Proportion	Topic Composition
79	9	0.12379	animale,sperimentazione,parte,senza,vivisezione,anni,scientifica,gennaio,due,oggi,solo,alternativi,dibattito,metodi,sempre,diritti,grazie,scientifici,meno,stato,
80	7	0.08127	non,perch,cosa,animalisti,fatto,caso,caterina,fare,essere,cos,invece,stata,altri,proprio,dire,dare,malattie,parlare,video,mai,
81	10	0.08018	non,animale,sperimentazione,ricerca,perch,animali,convegno,cattaneo,malattia,scientifico,ricercatori,vita,laboratorio,senatrice,direttiva,metodo,italiana,governo,elena,europea,
82	5	0.06774	animali,sperimentazione,ricerca,animale,salute,senato,metodi,rispetto,studi,alternative,de,incontro,alternativi,cos,diritto,sanit,test,cavie,commissione,legge,
83	18	0.06441	animali,test,sperimentazione,oggi,sperimentazioni,essere,prodotti,italiani,testare,punti,scientifica,medici,scopi,necessaria,accettabile,ipsos,cosmetici,opinione,molto,circa,
84	17	0.06414	pi,essere,solo,modo,vengono,vita,spesso,gi,ancora,prima,molti,realt,questione,senso,fare,capire,tratta,dunque,molte,esempio,
85	6	0.0593	pi,ricerca,persone,non,ricercatori,animali,solo,ora,sempre,malati,esempio,ancora,cos,anno,sotto,vita,tempo,altre,subito,cura,
86	8	0.05782	non,umani,specie,test,animali,metodi,perch,sperimentazione,milioni,viene,mai,dopo,vita,etica,campo,oltre,sviluppo,stesso,laboratorio,risposta,
87	15	0.05377	non,pu,scienza,qui,sel,credo,politica,animalista,forza,ragione,partito,interno,temi,fine,essere,parte,morte,etica,progresso,gruppo,
88	3	0.05349	direttiva,italia,commissione,ue,non,paese,gennaio,scientifici,recepimento,protezione,decreto,utilizzati,articolo,secondo,giorno,parlamento,euro,essere,legislativo,attuazione,
89	Super-topic,8[18566]	Proportion	Topic Composition
90	9	0.0956	animale,sperimentazione,parte,senza,vivisezione,anni,scientifica,gennaio,due,oggi,solo,alternativi,dibattito,metodi,sempre,diritti,grazie,scientifici,meno,stato,
91	4	0.06934	europea,animali,pi,legge,corte,fini,nazionale,ancora,normativa,stati,stato,parere,esperimenti,infatti,testo,europeo,gi,norme,diritto,ricercatori,
92	7	0.0498	non,perch,cosa,animalisti,fatto,caso,caterina,fare,essere,cos,invece,stata,altri,proprio,dire,dare,malattie,parlare,video,mai,
93	5	0.04932	animali,sperimentazione,ricerca,animale,salute,senato,metodi,rispetto,studi,alternative,de,incontro,alternativi,cos,diritto,sanit,test,cavie,commissione,legge,
94	1	0.04922	universit,milano,animale,proprio,simonsen,immagini,dati,dopo,alcuni,associazione,minacce,stati,italia,cento,altro,facebook,pubblico,padovan,sito,giorno,
95	17	0.04534	pi,essere,solo,modo,vengono,vita,spesso,gi,ancora,prima,molti,realt,questione,senso,fare,capire,tratta,dunque,molte,esempio,
96	12	0.04212	animali,non,uomo,farmaci,farmaco,modelli,pu,esseri,modello,umana,molto,risultati,topi,umano,ricerca,malattie,base,umane,cancro,altro,
97	6	0.03985	pi,ricerca,persone,non,ricercatori,animali,solo,ora,sempre,malati,esempio,ancora,cos,anno,sotto,vita,tempo,altre,subito,cura,
98	19	0.0383	animali,vivisezione,salute,lav,ministero,governo,schema,lorenzin,associazioni,art,presidente,affari,ministro,legislativo,delegazione,senza,solo,alternativi,tavolo,decreto,
99	18	0.03084	animali,test,sperimentazione,oggi,sperimentazioni,essere,prodotti,italiani,testare,punti,scientifica,medici,scopi,necessaria,accettabile,ipsos,cosmetici,opinione,molto,circa,
100	Super-topic,9[20390]	Proportion	Topic Composition
101	0	0.00501	the,of,and,to,for,that,or,human,is,it,have,page,we,by,models,are,full,research,testing,as,
102	3	0.0041	direttiva,italia,commissione,ue,non,paese,gennaio,scientifici,recepimento,protezione,decreto,utilizzati,articolo,secondo,giorno,parlamento,euro,essere,legislativo,attuazione,
103	5	0.0026	animali,sperimentazione,ricerca,animale,salute,senato,metodi,rispetto,studi,alternative,de,incontro,alternativi,cos,diritto,sanit,test,cavie,commissione,legge,
104	17	0.00225	pi,essere,solo,modo,vengono,vita,spesso,gi,ancora,prima,molti,realt,questione,senso,fare,capire,tratta,dunque,molte,esempio,
105	19	0.00225	animali,vivisezione,salute,lav,ministero,governo,schema,lorenzin,associazioni,art,presidente,affari,ministro,legislativo,delegazione,senza,solo,alternativi,tavolo,decreto,
106	4	0.00217	europea,animali,pi,legge,corte,fini,nazionale,ancora,normativa,stati,stato,parere,esperimenti,infatti,testo,europeo,gi,norme,diritto,ricercatori,
107	13	0.00203	false,name,lsdexception,locked,priority,semihidden,unhidewhenused,accent,twitter,medium,pi,cookie,list,grid,shading,offrirti,personalizzare,usando,web,light,
108	18	0.00196	animali,test,sperimentazione,oggi,sperimentazioni,essere,prodotti,italiani,testare,punti,scientifica,medici,scopi,necessaria,accettabile,ipsos,cosmetici,opinione,molto,circa,
109	7	0.00189	non,perch,cosa,animalisti,fatto,caso,caterina,fare,essere,cos,invece,stata,altri,proprio,dire,dare,malattie,parlare,video,mai,
110	12	0.00161	animali,non,uomo,farmaci,farmaco,modelli,pu,esseri,modello,umana,molto,risultati,topi,umano,ricerca,malattie,base,umane,cancro,altro

Tabella 4.3: Super-topic relativi al modello PAM

Conclusioni

Siamo partiti da un dataset di tweet riguardanti un singolo argomento, la sperimentazione animale, utilizzandoli in prima battuta per semplici analisi descrittive, individuando ad esempio gli utenti piú attivi, i momenti di maggior intensitá della discussione, le parole e gli n-grammi piú frequenti. Le analisi hanno peró sottolineato l'importanza dei retweet, come contenuto informativo, in un dataset poco polarizzato come lo era quello preso in considerazione. Abbiamo introdotto quindi una nuova funzione, RTHound, per l'individuazione e la clusterizzazione degli stessi, basata sulla distanza (di Levenshtein) tra questi. Non possiamo certo considerarci soddisfatti, le informazioni da scoprire sono ancora molte e analisi cosí semplici ci sono solo parzialmente d'aiuto; sono stati presentati a questo scopo modelli piú complessi per il text mining, quali LDA, PAM e le loro varianti gerarchiche, applicandoli prima ai tweet e successivamente ad un corpus di articoli. Attraverso un approccio bayesiano, questi modelli individuano un insieme di topic che riassuma al meglio il contenuto del corpus.

Confrontando i risultati di LDA e PAM, la capacitá di individuare topic specifici é sicuramente migliore nel secondo: come spiegato in dettaglio nel capitolo 4, il Pachinko Allocation ha dalla sua una flessibilitá nel gestire i topic che l'LDA non raggiunge. L'introduzione di livelli intermedi quali i super-topic, permette di considerare anche i topic come facenti parte di un vocabolario, lasciando libera la possibilitá che esista correlazione tra essi. I livelli intermedi introducibili sono teoricamente infiniti, anche se possiamo vedere che giá con un solo livello di super-topic il modello lavora egregiamente; in tabella 4.2 possiamo vederne i risultati.

Possiamo ritenerci abbastanza soddisfatti delle analisi svolte, l'informazione che si puó trarre é ampia e soprattutto specifica e fornisce certamente un'idea ben delineata degli argomenti trattati. Certo sarebbe interessante spingersi ancora oltre, confrontando i risultati delle varianti gerarchiche di LDA e PAM: questi ultimi sono modelli sicuramente piú potenti, capaci di rilassare ancora le assunzioni di base. Certo anche l'implementazione é piú faticosa ed esula dagli scopi di questo lavoro. Lo stesso ragionamento va fatto anche per il modello Twitter-Network, molto interessante dal punto di vista teorico: assumendo come variabili hashtag e autori oltre alle singole parole le capacitá predittive potrebbero essere notevoli. Si é visto infatti nelle analisi LDA dei tweet, pur essendo questo uno strumento potente, che il numero ridotto di parole ma soprattutto la mancanza nella gran parte dei casi di una mistura di argomenti nel singolo tweet, influiscano negativamente sui risultati.

Questa lunga e moderatamente vivace dissertazione si conclude, lasciando al lettore/lettrice lo spunto per lavori futuri e un po' di curiositá verso l'argomento Topic Model.

Appendice A

Stima dei parametri

Rendiamo piú esplicito il processo inferenziale alla base dei modelli descritti nel corso della tesi. Le variabili di interesse in questi modelli sono ϕ, la distribuzione dei topic sulle parole, e θ, la distribuzione dei topic sui documenti. Per la stima dei parametri sono stati proposti vari metodi, tra cui l'*Expectation-Maximization* (EM) e il *campionamento di Gibbs*. Il metodo EM ha difficoltà nel gestire i massimi locali della funzione di verosimiglianza; questo ha portato a ricercare metodi alternativi. In seguito verrà presentato il campionamento di Gibbs.

A.1 Campionamento di Gibbs

Il *campionamento di Gibbs*, (*Gibbs Sampler*), introdotto da Geman and Geman (Geman and Geman, 1984), è un caso particolare del *campionamento di Metropolis-Hastings* dove il valore candidato è sempre accettato e quindi $\alpha = 1$. Il punto di forza del Gibbs Sampler è che considera solamente distribuzioni condizionate univariate cioè la distribuzione dove tutte le variabili casuali tranne una sono fissate. Tali distribuzioni condizionate sono più facili da simulare rispetto alle più complesse distribuzioni congiunte e spesso hanno una forma semplice. Si simulano sequenzialmente n valori casuali dalle n distribuzioni condizionate a tutte le variabili tranne quella considerata piuttosto che generare un unico vettore $n-$dimensionale in un unico passo usando la distribuzione congiunta di tutte le n variabili. Per introdurre il Gibbs Sampler viene utilizzata una distribuzione casuale bivariata $p(x, y)$, e si supponga di riuscire a calcolare una o entrambe le marginali, $p(x)$ e $p(y)$. L'idea dell'algoritmo si basa sul principio che è più facile considerare una sequenza di distribuzioni considerate, $p(x|y)$ e $p(y|x)$. Le distribuzioni marginali si possono ottenere integrando la distribuzione congiunta $p(x, y)$, ad esempio $p(x) = \int p(x, y)\, dy$. L'algoritmo parte con alcuni valori iniziali per le due variabili casuali y_0 per y mentre x_0 viene generato dalla distribuzione condizionata $p(x|y = y_0)$. L'algoritmo poi usa x_0 per generare un nuovo valore y_1, estraendolo dalla distribuzione condizionata sul valore x_0, $p(y|x = x_0)$. Il Gibbs Sampler procede in questo modo:

$$x_i \sim p(x|y = y_{i-1})$$
$$y_i \sim p(y|x = x_i)$$

Ripetendo questo processo k volte, si ottengono k vettori bidimensionali in cui ogni dimensione corrisponde ad una generazione della relativa distribuzione condizionata. I punti così ottenuti o un loro sottoinsieme, (x_j, y_j) per $1 \leq j \leq m < k$, possono essere considerarati come valori simulati dalla distribuzione congiunta di tutte le variabili, dove m rappresenta il numero totale di campioni che si vogliono estrarre dalla distribuzione obbiettivo. Un iterazione di tutte le variabili univariate è spesso chiamata scan del sampler. Prima di ottenere m campioni è necessario:

1. iterare l'algoritmo per un sufficiente numero di volte (burn-in) per eliminare gli effetti della scelta dei valori iniziali;

2. poiché si vogliono m osservazioni i.i.d. dalla distribuzione obiettivo è pratica comune generarne $n \times m$ e successivamente collezionare una sola osservazione ogni n: tra due osservazioni ce ne saranno quindi n scartate. Questa tecnica viene utilizzata per ridurre la correlazione fra osservazioni poiché l'algoritmo genera una catena di Markov i cui valori sono strutturalmente dipendenti.

La sequenza del Gibbs converge alla distribuzione di equilibrio che è indipendente dai valori iniziali, e per costruzione questa distribuzione stazionaria è la distribuzione obiettivo da cui si vuole simulare.

La struttura del Gibbs Sampling utilizzata è costituita principalmente da due livelli: il primo consiste nell'estrazione della variabile indicatrice del topic $t_{j,i}$ (tavolo nella metafora CRF) associata alla parola $x_{j,i}$, mentre il secondo campiona la variabile $k_{j,t}$ (pietanza nella metafora CRF) che caratterizza l'argomento $t_{j,i}$ individuato nel $j-$esimo documento. Vediamo ora uno schema concettuale, che potrebbe essere utile per fornire una visione d'insieme dell'algoritmo Collapsed Gibbs Sampler ricavato dalla metafora del CRF.

1. Per tutti i documenti e quindi per $\forall j \in \{1, ..., J\}$ si procede come segue:

   (a) Per ognuna delle parole contenute nel $j-$esimo documento identificate da $x_{i,j}$ con $i \in \{1, ..., N_j\}$ si ha che (la parola considerata si suppone sia l'ultima arrivata e non contribuirà in nessun modo nei conteggi o alle altre quantità utilizzate nel seguito):

      i. si estrae la variabile indicatrice $t_{j,i}$, e possono verificarsi le due seguenti situazioni:

         A. se si è estratto un t già utilizzato si definisce $t_{j,i} = t$. In questo caso non occorre campionare $k_{j,t}$ poiché questo viene ereditato già dal vecchio cluster;

         B. altrimenti si aggiorna il numero totale di cluster identificati all'interno del documento j$-$esimo, $m_{j,\cdot} = m_{j,\cdot} + 1$, e si definisce $t_{j,i} = m_{j,\cdot}$. Questo nuovo cluster dovrà essere caratterizzato estraendo anche l'indicatore $k_{j,t}$;

   (b) una volta aggiornati tutti gli indicatori $t_{j,i}$ e aggiornate tutte le variabili conteggio necessarie per caratterizzare le distribuzioni condizionate degli indicatori, si procede con il campionamento degli indicatori $k_{j,t}$; per i valori unici di $t_{j,i}$ definiti all'interno del

documento $j-$esimo (analogamente al caso precedente il cluster considerato si suppone l'ultimo arrivato e non contribuirà in nessun modo nei conteggi o alle altre quantità utilizzate nel seguito. Non verranno considerate quindi tutte le parole associate a quel cluster):

i. si estrae l'indicatore $k_{j,t}$ utilizzando la distribuzione definita in (4.5.5) e possono verificarsi due situazioni:
   A. se si è estratto un k già utilizzato si definisce $k_{j,i} = k$;
   B. altrimenti si aggiorna il numero globale di argomenti rappresentati esplicitamente $K = K + 1$ e si definisce $k_{j,t} = K + 1$;

Riprendendo la metafora del CRF grazie allo schema proposto in precedenza risulta essere più chiara la struttura a più livelli che caratterizza il processo: al primo livello (estrazione dei t) si considerano le parole come i clienti, mentre al secondo livello (campionamento dei k) sono gli stessi cluster ad essere assimilati ai clienti.
Griffiths and Steyvers (Landauer, McNamara, Dennis, and Kintsch, 2004) hanno dimostrato che la distribuzione condizionata può essere calcolata come segue:

$$\Pr(z_i = j|\mathbf{z}_{-i}, w_i, d_i, \cdot) \propto \frac{C^{WT}_{w_ij} + \beta}{\sum_{w=1}^{W} C^{WT}_{wj} + W\beta} \frac{C^{DT}_{d_ij} + \alpha}{\sum_{t=1}^{T} C^{DT}_{d_it} + T\alpha}$$

Dove C^{WT} e C^{DT} sono matrici di conteggi di dimensioni $W \times T$ e $D \times T$ rispettivamente; C^{WT} contiene il numero di volte che la parola w è assegnata al topic j, senza includere il passo corrente i, mentre C^{DT} contiene il numero di volte che il topic j è assegnato al documento d, senza includere il passo corrente i. La parte sinistra dell'equazione rappresenta la probabilità della parola w sul topic j, la parte destra la probabilità del topic j sul documento d. Ogni volta che una parola è assegnata al topic j, la probabilità di assegnare altre parole specifiche a questo topic aumenta. Allo stesso tempo, se il topic j è usato più volte nello stesso documento, aumenta la probabilità che le parole del documento vengano assegnate ad esso. Quindi le parole sono assegnate ai topic più verosimili come ai topic predominanti in un documento.
L'algoritmo fornisce stime dirette di z per ogni parola. Spesso però sono le

stime di ϕ' e θ' ad interessare. Queste possono essere ottenute come segue:

$$\phi_i'^{(d)} = \frac{C_{ij}^{WT} + \beta}{\sum_{k=1}^{W} C_{kj}^{WT} + W\beta} \qquad \theta_j'^{(d)} = \frac{C_{dj}^{DT} + \alpha}{\sum_{k=1}^{T} C_{dk}^{DT} + T\alpha}$$

I valori corrispondono rispettivamente alla ditribuzione predetta relativa all'estrazione di una nuova parola i dal topic j, e all'estrazione di un nuovo topic j nel documento d, e sono inoltre le medie a posteriori di queste quantità condizionate ad un particolare valore di z.

Appendice B

Codici

I grafici presenti nella tesi sono disponibili con il pacchetto `ggplot2`.Di seguito sono presenti tutti i relativi codici, mentre per semplicitá nel testo sono stati lasciati quelli disponibili con i pacchetti di base in `R`.

```
require(ggplot2)
ggplot()+geom_bar(aes(x=na.omit(tw$rt)))+
        theme(axis.text.x=element_text(angle=-90,size=6))
        +xlab(NULL)

plot(table(tw$created.hours),ylab="tweets")
ggplot(tw,aes(created.hours))
        +geom_area(stat = "bin", binwidth=1, drop=TRUE)

plot(table(tw$created.day),ylab="tweets")
ggplot(tw,aes(created.day))
        +geom_area(stat="bin",binwidth=1,drop=T)

user.date=data.frame(tw$screenName,tw$created.day)
orianoPER=as.data.frame(subset
        (user.date,user.date[,1]=="orianoPER"))

plot(table(orianoPER[,2]))
ggplot(orianoPER,aes(as.factor(tw.created.day)))
        +geom_bar(stat="bin",binwidth=1,drop=T)
        +theme(axis.text.x=element_text(angle=-90,size=10))
```

```
ggplot(top.date, aes(Data,fill=Retweet,group=Retweet))
        + geom_bar()+theme(axis.text.x=element_text(angle=-90,size=6))
```

Appendice C

Itastopword

	1	2	3	4
1	abbia	é	ne	stavamo
2	abbiamo	ebbe	ne	stavano
3	abbiano	ebbero	né	stavate
4	abbiate	ebbi	negl	stavi
5	ad	ecc	negli	stavo
6	agl	ed	nei	stemmo
7	agli	era	nel	stesse
8	ai	erano	nell	stessero
9	al	eravamo	nella	stessi
10	all	eravate	nelle	stessimo
11	alla	eri	nello	steste
12	alle	ero	noi	stesti
13	allo	essendo	nostra	stette
14	anche	etc	nostre	stettero
15	anziche	fa	nostri	stetti
16	anziché	faccia	nostro	stia
17	avemmo	facciamo	ogni	stiamo
18	avendo	facciano	per	stiano
19	avesse	facciate	perche	stiate
20	avessero	faccio	perché	sto
21	avessi	facemmo	peró	su
22	avessimo	facendo	po	sua
23	aveste	facesse	pó	sue
24	avesti	facessero	poi	sugl
25	avete	facessi	puó	sugli
26	aveva	facessimo	qual	sui
27	avevamo	faceste	quale	sul
28	avevano	facesti	quali	sull
29	avevate	faceva	quando	sulla

	1	2	3	4
30	avevi	facevamo	quanta	sulle
31	avevo	facevano	quante	sullo
32	avrá	facevate	quanti	suo
33	avrai	facevi	quanto	suoi
34	avranno	facevo	quell	ti
35	avrebbe	fai	quella	tra
36	avrebbero	fanno	quelle	tu
37	avrei	fará	quelli	tua
38	avremmo	farai	quello	tue
39	avremo	faranno	quest	tuo
40	avreste	farebbe	questa	tuoi
41	avresti	farebbero	queste	tutti
42	avrete	farei	questi	tutto
43	avró	faremmo	questo	un
44	avuta	faremo	sa	una
45	avute	fareste	sará	uno
46	avuti	faresti	sarai	vabbé
47	avuto	farete	saranno	vi
48	che	faró	sarebbe	via
49	chi	fece	sarebbero	voi
50	chissá	fecero	sarei	vostra
51	ci	feci	saremmo	vostre
52	ció	fosse	saremo	vostri
53	cmq	fossero	sareste	vostro
54	coi	fossi	saresti	xche
55	col	fossimo	sarete	xché
56	come	foste	saró	xké
57	comunque	fosti	se	a
58	con	fu	sei	b
59	contro	fui	si	c
60	cose	fummo	sia	d
61	cosí	furono	siamo	e
62	cui	giá	siano	f
63	da	gli	siate	g
64	dá	ha	siete	h
65	dagl	hai	sono	i
66	dagli	hanno	st	j
67	dai	ho	sta	k
68	dal	il	stai	l
69	dall	in	stando	m
70	dalla	io	stanno	n
71	dalle	la	stará	o
72	dallo	le	starai	p
73	degl	lei	staranno	q
74	degli	li	starebbe	r
75	dei	lo	starebbero	s
76	del	loro	starei	t
77	dell	lui	staremmo	u
78	della	ma	staremo	v
79	delle	mi	stareste	w
80	dello	mia	staresti	x
81	di	mie	starete	y
82	dov	miei	staró	z
83	dove	mio	stava	

Bibliografia

Mario Annau. *tm.plugin.webmining: Retrieve structured, textual data from various web sources*, 2012. URL `http://CRAN.R-project.org/package=tm.plugin.webmining`. R package version 0.9.

Mario Annau. *boilerpipeR: Interface to the boilerpipe Java library by Christian Kohlschutter (http://code.google.com/p/boilerpipe/)*, 2014. URL `http://CRAN.R-project.org/package=boilerpipeR`. R package version 1.1.

Blei, D. M., Ng, A. Y., Jordan, and M. I. Latent dirichlet allocation. *Journal of Machine Learning Research, 3, 993-1022*, 2003.

D. Blei, D. Mimno, Griffiths, T.L., Jordan, M.I., Tenenbaum, and J. B. Hierarchical topic models and the nested chinese restaurant process. *Advances in Neural Information Processing Systems 16, Cambridge, MA, MIT Press*, 2004.

Christian Buchta, Kurt Hornik, Ingo Feinerer, and David Meyer. *tau: Text Analysis Utilities*, 2012. URL `http://CRAN.R-project.org/package=tau`. R package version 0.0-15.

Jonathan Chang. *lda: Collapsed Gibbs sampling methods for topic models.*, 2012. URL `http://CRAN.R-project.org/package=lda`. R package version 1.3.2.

Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006. URL `http://igraph.org`.

Mimno David, Li Wei, and McCallum Andrew. Mixtures of hierarchical topics with pachinko allocation. *Proceedings of the* 24^{th} *International Conference on Machine Learning*, 2007.

Ian Fellows. *wordcloud: Word Clouds*, 2013. URL `http://CRAN.R-project.org/package=wordcloud`. R package version 2.4.

Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.1984.4767596.

Jeff Gentry. *twitteR: R based Twitter client*, 2013. URL `http://CRAN.R-project.org/package=twitteR`. R package version 1.1.7.

Kurt Hornik, Patrick Mair, Johannes Rauch, Wilhelm Geiger, Christian Buchta, and Ingo Feinerer. The textcat package for n-gram based text categorization in R. *Journal of Statistical Software*, 52(6):1–17, 2013. URL `http://www.jstatsoft.org/v52/i06/`.

T. Landauer, D. McNamara, S. Dennis, and W. Kintsch. Latent semantic analysis: A road to meaning. *Laurence Erlbaum, Probabilistic Topic Models*, 2004.

Kar Wai Lim, Changyou Chen, and Wray Buntine. Twitter-network topic model: A full bayesian treatment for social network and text modeling. *NIPS2013 Topic Model workshop*, December 2013.

Andrew Kachites McCallum. Mallet: A machine learning for language toolkit, 2002. http://mallet.cs.umass.edu.

Matteo Redaelli. Matteo redaelli blog, 2014. http://www.redaelli.org/matteo-blog/.

Dario Solari, Livio Finos, Matteo Redaelli, con contributi di Marco Rinaldo, Maddalena Branca, and Federico Ferraccioli. *TextWiller: Collection of functions for text mining, specially devoted to the italian language*, 2013. R package version 1.0.

L. Tierney. Markov chains for exploring posterior distributions. *Annals of Statististics*, 1994.

Li Wei and Blei Davidand McCallum Andrew. Nonparametric bayes pachinko allocation. *CoRR*, 2007a.

Li Wei and McCallum Andrew. Pachinko allocation: Dag-structured mixture models of topic correlations. *Proceedings of the* 23^{rd} *International Conference on Machine Learning*, 2007b.