CLUSTERING CON GIBBSLDA++
LATENT SEMANTICS INDEXING

ILARIA CEPPA
MARCO GRANDI
MARCO PONZA

Relazione per il progetto del corso
di Information Retrieval

INDICE

1 INTRODUZIONE

1.1 OBIETTIVO DEL PROGETTO

L'obiettivo del progetto è quello di realizzare, sperimentare ed analizzare i risultati di un'applicazione, scritta in C/C++, che, usando la libreria GibbsLDA++ [3], generi cluster di news, ricevendo come input dataset di news in italiano.

I dataset si differenziano tra loro per il topic principale di cui parlano le rispettive news; sono quindi presenti dataset su: economia, italia, mondo, sport, spettacolo e scienza e tecnologia.

In questo capitolo viene ora fatta una breve panoramica su:

- Realizzazione e sperimentazione (Sezione 1.2);
- Analisi dei risultati ottenuti (Sezione 1.3);
- GibbsLDA++ (Sezione 1.4)

Per maggiori dettagli sui primi due punti il lettore è rimandato ai relativi capitoli.

1.2 REALIZZAZIONE E SPERIMENTAZIONE

L'applicazione è stata realizzata effettuando delle scelte progettuali discusse nel Capitolo 2, sperimentando e valutando diverse tecniche di:

- Preprocessing, allo scopo di ottenere degli input per la libreria filtrati da quei termini considerati poco importanti;
- Postprocessing, allo scopo di rimuovere outlier e di validare i cluster;

al fine di generare, con l'utilizzo della libreria GibbsLDA++, dei cluster che fossero coesi e paragonabili ai cluster fornitici in partenza.

1.3 ANALISI

La fase di analisi, descritta nel Capitolo 3, è stata svolta effettuando un confronto "a occhio" tra i migliori risultati che sono stati ottenuti nella fase di sperimentazione e i clustering forniti per il confronto. Per la valutazione sulla bontà dei clustering è stato utilizzato un apposito tool per la visualizzazione dei cluster tramite browser.

1.4 GIBBSLDA++: LA LIBRERIA

La libreria GibbsLDA++ è un'implementazione, scritta in C/C++, del Latent Dirichlet Allocation (LDA) usando tecniche di Gibbs Sampling. [3]

In breve, sia LDA che LSI descrivono modelli matematici che possono essere usati nell'ambito dell'information retrieval, ma presentano alcune differenze:

- LSI esamina le parole di un documento e cerca le loro relazioni con altre parole, presentando, però, la debolezza dell'ambiguità. Per esempio, data la parola *office*, non è possibile sapere se si sta parlando di *Microsoft Office* o di *office* inteso come l'ufficio. [2]
- LDA, invece, è un estensione dell'LSI. Le parole, infatti, vengono raggruppate in topic: una stessa parola può, quindi, comparire in più topic. LDA, quindi, risolve il problema dell'ambiguità confrontando un documento con più topic e determinando quale topic è più inerente a quel documento. [2]

Gibbs Sampling, invece, è un algoritmo Markov Chain Monte Carlo (MCMC) che permette di approssimare una distribuzione di probabilità specificata. Per ulteriori informazioni, si rimanda a [1].

1.4.1 *Parametri*

Per usare la libreria è necessario specificare il modo di utilizzo e il file di input. Per il nostro scopo, ovvero quello di calcolare il modello LDA da zero, bisogna utilizzare l'opzione `-est`.
Ulteriori parametri che è possibile specificare sono:

- `ntopics`: numero di topic.
- `niters`: numero di iterazioni.
- `twords`: numero di parole più probabili per ogni topic. Questo parametro non influenza il risultato generato dalla libreria.

1.4.2 *Formattazione dell'Input*

La libreria prende in input dataset con il seguente formato:

```
[M]
[document1]

[document2]

...
```

```
[documentM]
```

Dove `M` è il numero totale di documenti e il documento i-esimo consiste in una lista di `N` parole[1]:

```
[i-document] = [i1-word] [i2-word] ... [iN-word]
```

1.4.3 *Output*

La libreria genera diversi file di output, dei quali quelli con la seguente estensione sono utili al nostro scopo:

- `theta`: file contenente le distribuzioni topic-documento. Ogni riga di questo file rappresenta un documento e ogni colonna rappresenta la probabilità che un dato documento appartenga ad un certo topic.
- `twords`: file che contiene le parole più probabili per un certo topic.

1 Il valore di `N`, chiaramente, può essere diverso a seconda del documento.

2 SCELTE PROGETTUALI

La scelta che è stata fatta è quella di sviluppare un'applicazione unica il cui core fosse la libreria GibbsLDA++ e che permettesse il clustering di dataset conformi al formato del progetto.
Il processo di clustering viene completamente gestito dalla libreria, quindi durante lo sviluppo si è posta enfasi soprattutto sulle fasi di preprocessing e postprocessing. Sono state provate diverse tecniche, per poi scegliere e includere nel progetto quelle che sembravano dare i risultati migliori.

2.1 PREPROCESSING

Il preprocessing consiste nella manipolazione e nel filtraggio del testo[1] che costituisce ogni notizia. I metodi sviluppati sono:

- *Rimozione della punteggiatura.* Si limita a rimuovere la punteggiatura presente nel testo della notizia.
- *Rimozione delle stopword e delle parole molto frequenti.* L'algoritmo utilizzato per il clustering si basa fortemente sulla presenza delle stesse parole nei documenti per stabilirne la somiglianza. L'eliminazione delle stopword ha contribuito notevolmente al miglioramento del clustering ottenuto; un ulteriore miglioramento si è avuto osservando che nei risultati erano ancora presenti parole molto comuni, che non erano stopword, e che disturbavano molto la creazione dei cluster: queste parole sono state aggiunte manualmente al file di stopword utilizzato dal programma per filtrare il testo delle notizie.
- *Rimozione delle parole troppo corte.* Vengono rimosse le parole che hanno meno di 3 caratteri, perché non sono solitamente interessanti per il clustering.
- *Creazione di shingle.* Le parole in ogni documento vengono sostituite con shingle, ottenuti concatenando due o più parole successive. Il particolare caso di shingle formati da due parole ha migliorato il clustering, permettendo di ottenere cluster fra i cui termini importanti appaiono coppie di parole con un preciso significato. Per esempio il cluster nelle cui notizie appare lo shingle costituito da nome e cognome di una persona.
- *Rimozione delle parole tramite IDF.* Sempre per cercare di eliminare parole molto comuni, presenti in molte news e che quindi

1 Tale testo è il risultato della concatenazione del titolo al corpo della notizia.

disturbano il clustering, si è implementato un filtro che elimina le parole in base alla loro *Inverse Document Frequency*. Durante la fase di caricamento delle news viene calcolato per ogni parola t il numero di documenti in cui in questa occorre, N_t, dopodichè tramite la formula $\texttt{idf} = \ln \frac{N}{N_t}$ ad ogni termine viene associato il suo IDF. Vengono quindi eliminate tutte le parole[2] per cui $\texttt{idf} \leqslant 2.5$, ovvero parole troppo frequenti, oppure $\texttt{idf} \geqslant 7.5$, ovvero parole molto rare: queste vengono eliminate perché non utili ai fini del clustering; l'eliminazione dei termini che compaiono solo una o due volte riduce notevolmente la dimensionalità di ogni documento.[3]

Questa fase termina con la scrittura dei testi delle notizie sul file che rappresenta l'input della libreria, rispettando il formato richiesto da questa.

2.2 CREAZIONE DEI CLUSTER E POSTPROCESSING

Il clustering è ottenuto eseguendo la libreria e considerando due dei file prodotti in output, come specificato nella sezione 1.4. A partire da tali file vengono quindi ricostruiti i cluster che vengono sottoposti alla fase di postprocessing.
Dal file con estensione `twords` vengono recuperate le parole più probabili per ogni cluster con la relativa probabilità di appartenenza. Mentre dal file con estensione `theta` si ottiene il vettore di appartenenza ai cluster per ogni documento, ovvero la probabilità che un documento ha di appartenere a ogni cluster. Infine, ogni documento viene assegnato ad un cluster, scegliendo quello che nel vettore presenta la probabilità maggiore per quel documento.

Le informazioni estratte dai due file sono state utilizzate nelle fase di postprocessing per raffinare i cluster, eliminando i documenti non inerenti al topic. Le tecniche sviluppate sono:

- *Soglia sulla probabilità.* Si è osservato che molte news venivano associate ad un topic pur presentando una probabilità molto bassa di appartenervi: queste notizie risultavano in effetti sempre molto scollegate dalle altre del cluster, e si è quindi deciso di provare a eliminarle.

  Utilizzare una soglia fissa, ad esempio eliminando le news con una probabilità inferiore allo 0.1, non ha portato a dei buoni risultati: in alcuni casi venivano eliminate notizie che invece erano relative al topic, in altri casi notizie palesemente sbagliate non venivano cancellate.

2 Se i documenti sono stati precedentemente trasformati e contengono shingle, i termini su cui opera questo filtro sono gli shingle.

3 I valori di soglia sono stati scelti in maniera empirica.

Definire un unico valore soglia per la probabilità di appartenenza ad un cluster non è infatti possibile, e si è quindi provato a definire una soglia variabile: per ogni cluster vengono calcolate, su tutti i documenti in esso contenuti, la media (μ) e la deviazione standard (σ) della probabilità con cui un documento appartiene a quel cluster; un documento viene quindi eliminato dal cluster se la sua probabilità p è tale che $p < \mu - \sigma$.

- *Distanza dal centroide.* Per eliminare le notizie poco collegate ad un cluster si è anche provato a considerare le probabilità che legano un documento ad ognuno dei K cluster come le coordinate del documento in uno spazio K-dimensionale. Dato un cluster C_j, se ne calcola il centroide, un vettore c_j di K elementi tale che $c_j[i] = \frac{\sum_{p=0}^{M_j} d_p[i]}{M_j}$ dove $d_p[i]$ è la probabilità con cui il documento p-esimo del cluster j è legato al cluster i, e M_j rappresenta il numero di documenti che si vuole utilizzare per la costruzione del centroide: per evitare l'influenza degli outlier del cluster è infatti meglio non considerare tutti i documenti di un dato topic, ma solo i primi, i più importanti per il cluster.

  Si calcola poi successivamente la distanza[4] di tutti i documenti del cluster dal centroide, e vengono eliminati i documenti la cui distanza è maggiore di una certa soglia.

- *Rimozione basata sui termini che descrivono il topic.* Osservando i cluster ottenuti è emerso che molto spesso erano presenti notizie che non contenevano nessuna delle parole più probabili che caratterizzavano un cluster: queste notizie risultavano essere outlier, e si è quindi deciso di utilizzare questa informazione sulle parole caratterizzanti un topic per eliminare questi documenti.

  Dato il numero n di termini da considerare, ogni documento che non contiene nessuno dei primi n termini che descrivono il cluster a cui appartiene viene quindi rimosso dal cluster. Un soglia pari a 2 si è dimostrata essere un buon valore.

La fase di postprocessing, prima di produrre il risultato finale, stabilisce la validità di ogni cluster. Considerando anche i metodi di postprocessing provati, non è emerso un modo ottimale per calcolare la validità di un cluster. Si è quindi scelto di ritenere un cluster valido se la probabilità con cui il documento più probabile per quel cluster è maggiore o uguale della media con cui ogni documento è legato al proprio cluster.

2.3 SCELTA FINALE

Dopo aver provato diverse combinazioni dei metodi descritti in precedenza, si è selezionata quella che secondo i test effettuati permetteva

4 Sono state provate sia la distanza euclidea che la cosine similarity.

di ottenere risultati migliori.

Come prima cosa si è scelto di utilizzare i file contenenti le notizie stemmate: lo stemming dei vocaboli è infatti raccomandato dalla documentazione della libreria stessa, e in effetti i risultati delle prove mostravano che utilizzare documenti non stemmati rendeva molto più difficoltoso il processo di clustering.

Tutte le tecniche di preprocessing descritte si sono rivelate utili: si è quindi scelto di utilizzarle tutte, anche se la rimozione della punteggiatura non risulta necessaria per documenti stemmati.

Per quanto riguarda il postprocessing invece è emerso che l'eliminazione basata sui termini che descrivono il topic, l'ultima delle tecniche descritte, permetteva di ottenere risultati nettamente migliori rispetto alle altre due.La rimozione di documenti in base ad una soglia sulla probabilità oppure in base alla distanza dal centroide produceva dei cluster che erano di poco migliori rispetto a quelli che si sarebbero ottenuti senza effettuare nessun postprocessing sui risultati della libreria: pochi documenti venivano riconosciuti come non attinenti, e il risultato erano cluster molto grandi, con un rilevante numero di outlier. Per questo motivo si è deciso di applicare un'unica tecnica di postprocessing, quella che elimina i documenti che non contengono i termini più rilevanti del topic, che genera risultati di gran lunga migliori.

2.4 PARAMETRI DI DEFAULT ED OUTPUT

L'applicazione prodotta utilizza la combinazione indicata nella sezione precedente, fornendo per ogni metodo che lo richiede un parametro di default. Questi valori sono stati scelti in maniera euristica in base alla bontà del clustering ottenuto, valutata "a occhio". Per alcuni parametri, i valori preimpostati sono rilassati rispetto a quelli riportati nella sezione 3.1 che sono stati utilizzati per produrre di file clustering per il confronto.

L'applicazione utilizza tutti i filtri sviluppati per la fase di preprocessing e nello specifico crea shingle di due parole. Per quanto riguarda il clustering invece sono utilizzati i parametri di default della libreria per α e β[5]. Il numero di iterazioni fatte è impostato a 4000 e la creazione di modelli intermedi avviene ogni 2000 iterazioni. I parametri fondamentali per la libreria sono il numero di cluster, che di default vale 200, ed il numero di parole più probabili per cluster di cui tenere traccia, che è impostato a 5. Valori diversi per quest'ultimo parametro non modificano l'output della libreria, ma un valore maggiore riporterebbe in output parole con una probabilità di appartenenza molto bassa. Infine il parametro del postprocessing indica il numero di termini più importanti da considerare per l'eliminazione di un documento dal cluster. Di default tale valore è 3.

5 Rispettivamente pari alla divisione di 50 per il numero di cluster ed a 0.1.

I cluster prodotti in output vengono ulteriormente filtrati, per eliminare i cluster troppo piccoli che solitamente sono poco significativi. La dimensione minima di un cluster, ovvero il numero di documenti che gli appartengono, è impostata a 10. Da notare poi che i cluster vengo ordinati per dimensione decrescente a parità di validità e che i documenti in ogni cluster sono ordinati secondo valori decrescenti di probabilità di appartenenza. Anche i termini più probabili per ogni cluster sono ordinati secondo tale criterio.
Per sapere come modificare i parametri o disabilitare alcuni filtri, si rimanda al capitolo 5.

3 CONFRONTO TRA CLUSTERING

In questo capitolo si discutono le differenze tra il clustering ottenuto utilizzando la libreria GibbsLDA++ e quello fornito, per valutare se l'utilizzo della tecnica del Latent Semantic Indexing permette di ottenere risultati migliori.
Per effettuare questo confronto sono stati generati i file di clustering dei vari dataset, utilizzando i parametri descritti nella prossima sezione. Per ogni dataset si sono confrontati qualitativamente i due clustering, soffermandoci soprattutto sui primi cluster di entrambi i file, ritenuti i più interessanti.
Nelle prossime sezioni sono riportati nel dettaglio i risultati di questo confronto su tutti i sei dataset.

3.1 PARAMETRI UTILIZZATI

Per fare il confronto tra i cluster generati mediante LSI e i cluster forniti, sono stati svolti diversi test, al fine di trovare la combinazione di parametri che generasse i cluster migliori.
I cluster usati per il confronto sono stati generati utilizzando i seguenti parametri:

- numero di cluster = 200;
- numero di termini = 3, ovvero i termini di cui si è tenuto traccia;
- iterazioni = 4000. Il numero di iterazioni è stato fissato a 4000 in quanto:
  - Con questo numero di iterazioni i cluster generati sono molto buoni;
  - Il tempo impiegato per generare i cluster con un numero maggiore di iterazioni è troppo elevato;
- $\alpha = 50/200 = 0.25$ (dove 200 è il numero di cluster);
- $\beta = 0.1$;
- nella fase di preprocessing si sono utilizzati tutti i filtri: punteggiatura, stopword, creazione di shingle e idf;
- parole per shingle = 2, ovvero uno shingle è formato dalla concatenazione di due parole successive;
- numero di termini per l'eliminazione = 2, ovvero il filtro sui documenti di un cluster considerava solo i primi 2 termini dei tre di cui si teneva traccia;

- sono stati prodotti in output tutti i cluster, senza utilizzare la soglia sulla dimensione.

3.2 ECONOMIA

Questo dataset, assieme a Scienza e Tecnologia, è risultato essere uno dei più difficili da clusterizzare: le notizie relative a questo argomento utilizzano infatti spesso molte parole comuni (come crescita trimestrale, profitti, fatturato, industria, ecc...) nonostante gli argomenti trattati siano in realtà diversi.

Un esempio di questo lo troviamo nel secondo cluster del file `economia_0`, che contiene notizie riguardanti molte aziende, diverse tra loro; le notizie sono accomunate dal fatto che riportano i profitti di queste aziende nel corso del terzo trimestre dell'anno: in pratica una qualsiasi notizia contenente le parole crescita trimestrale, utili, trimestre si ritrova in questo cluster, che non risulta quindi molto interessante. Un cluster simile è presente anche nel clustering con LSI in quarta posizione.

Nel file `economia_0` troviamo un primo cluster contenente ben 372 news: il cluster contiene notizie sull'andamento dello spread, ma anche notizie sugli andamenti delle borse europee, americane e giapponesi; nel nostro clustering invece sono presenti due gruppi di notizie, i primi due, contenententi rispettivamente 179 e 146 notizie: il primo raccoglie le notizie sull'andamento delle borse, mentre il secondo contiene solamente notizie riguardanti lo spread.

Anche i successivi cluster di `economia_0` risultano poco coesi: si trovano riunite in un unico gruppo notizie riguardanti uno sciopero della CGIL a Roma e una sentenza riguardante la Fiat, e in un altro gruppo news sull'Alitalia e news su Intesa San Paolo. Nel clustering generato con LSI invece sono presenti:

- un cluster relativo unicamente alla sentenza della Fiat, con pochissimi outlier;
- uno sullo sciopero della CGIL che contiene anche notizie relative a dichiarazioni fatte dal segretario generale della CGIL durante questo sciopero, quindi comunque collegate all'argomento principale, e notizie su dichiarazioni del ministro dell'economia, che invece non sembrano molto connesse alle altre;
- un cluster sull'Alitalia, che contiene in gran maggioranza notizie attinenti all'argomento, nonstante siano presenti alcuni outlier;
- un cluster su Intesa San Paolo che contiene quasi esclusivamente news relative al gruppo bancario.

Altri cluster riescono invece a catturare notizie relative ad unico argomento anche in `economia_0`: ad esempio è presente un cluster con

notizie sul Cancelliere tedesco Merkel, uno riguardante il ministro dell'economia Grilli e la legge di stabilità, uno relativo all'agenzia di rating americana Moody's. Questi cluster sono presenti anche nel clustering con LSI con all'incirca lo stesso numero di news.

Dal confronto sui cluster di questo dataset è quindi emerso che il clustering generato tramite il Latent Semantic Indexing riesce ad ottenere risultati migliori: i cluster sono infatti più coesi, con notizie in maggioranza relative ad un unico argomento. Comunque, a causa della difficoltà di questo tipo di notizie, i cluster ottenuti non sono perfetti, ma risultano sempre contenere un certo numero di news non attinenti al topic, oppure contenere diverse notizie su entità diverse accomunate dall'uso comune di molte altre parole.

3.3 ITALIA

Anche per questo dataset il clustering ottenuto con LSI è interessante. In questo caso solo i primi cluster (20-30) sono significativi, risultando molto coesi tra loro e relativi agli stessi topic presenti nei primi cluster del file `italia_0`.
Più nel dettaglio:

- Alcuni dei cluster presenti nel file fornito, come ad esempio quello relativo ad un omicidio avvenuto a Palermo, al processo di Salvatore Parolisi e altri, sono individuati anche dalla libreria, con quasi lo stesso numero di notizie.

- Alcuni cluster, come quello relativo a notizie sulla CGIL e quello relativo all'indagine su Sandro Sisler, sono individuati anche nel clustering effettuato ma sono leggermente più grandi, e alcune delle notizie (comunque sempre tra le ultime del cluster) non sono attinenti al tema.

  In altri casi invece dei cluster individuati dal Latent Semantic Indexing sono più grandi di quelli presenti nel file `italia_0`, perchè più generali. Ad esempio nel nostro clustering è presente un unico gruppo di notizie riguardanti Berlusconi, mentre nel clustering fornito è presente un cluster riguardante solo il processo Ruby.

- Alcuni cluster individuati usando il Latent Semantic Indexing non sono presenti nel file fornito: è il caso ad esempio di un cluster contenente notizie riguardante Maroni, Formigoni e le elezioni in Lombardia, e di un cluster relativo al processo per il naufragio della nave Costa Concordia.

In conclusione, anche per questo dataset, come per il precedente, possiamo dire che in generale la qualità dei cluster ottenuti è paragonabile a quella del clustering utilizzato per il confronto; la libreria

utilizzata individua infatti quasi gli stessi gruppi di notizie, e in alcuni casi ne individua alcuni interessanti che non sono presenti in `italia_0`.

3.4 MONDO

Su questo dataset il clustering ottenuto con LSI è molto buono, almeno per quanto riguarda i primi venti cluster, anche confrontato con il clustering del file `mondo_0`: tutti i cluster (tranne alcune eccezioni), contengono news molto coese tra loro.
Facendo un'analisi un po' più dettagliata sui primi 10-20 cluster, possiamo affermare che:

- I cluster generati con LSI hanno, in generale, dimensioni inferiori rispetto ai cluster di `mondo_0`. Per esempio:
  - I rispettivi cluster su Obama, entrambi nella prima posizione dei due clustering, presentano 182 news in `mondo_0`, mentre 166 news nel cluster LSI.
  - Il cluster sugli scontri in Grecia, presente in seconda posizione in `mondo_0`, possiede 103 news, mentre il rispettivo cluster, generato con LSI, si trova in sesta posizione e possiede 51 news.
  - Il cluster sulle sanzioni contro l'Iran, presente in terza posizione in `mondo_0`, possiede 69 news, mentre il rispettivo cluster generato con LSI si trova in seconda posizione e possiede 86 news che, però, non sono così ben coese: quest'ultime, infatti, parlano, oltre che delle sanzioni contro l'Iran, anche di argomenti politici su Siria e Iraq.
- I cluster generati dal Latent Semantic Indexing si trovano in posizioni molto simili a quelli presenti in `mondo_0`:
  - Il cluster su Obama si trova in entrambi i clustering in prima posizione;
  - Il cluster su Cuba si trova in `mondo_0` in quinta posizione, mentre nel clustering generato con LSI, in settima posizione;
  - Il cluster su Felix Baumgartner si trova in `mondo_0` in sesta posizione, mentre nel clustering generato con LSI, in ottava posizione;
  - Il cluster sul referendum per l'indipendenza della Scozia si trova in `mondo_0` in settima posizione, mentre nel clustering generato con LSI, in nona posizione;
- Alcuni cluster presenti in `mondo_0` sono migliori di quelli del clustering generato con LSI e viceversa. Per esempio:

  - Il cluster sull'attentato in Libano, presente in `mondo_0`, contiene news molto coese tra loro, mentre un cluster simile, generato da LSI, contiene anche news che parlano delle dimissioni del premier libanese;
  - Il cluster sulla Siria è presente nelle prime posizioni nel clustering generato con LSI mentre, nel file `mondo_0`, è presente un cluster simile che, però, si può trovare solo attorno alla venticinquesima posizione e con dimensioni molto inferiori rispetto a quello generato da LSI;
  - Il cluster sulla meningite è presente solo in `mondo_0`;
  - Il cluster sulle Pussy Riot è presente solo nel clustering generato con LSI.

In conclusione, su questo dataset, il clustering generato con LSI è paragonabile con quello di `mondo_0`, in quanto vengono prodotti, nella maggior parte dei casi, dei cluster molto simili, ma con dimensioni inferiori, a quelli forniti. Inoltre, sebbene LSI contenga dei cluster non presenti in `mondo_0`, alcuni cluster che si possono trovare in quest'ultimo non vengono individuati.

3.5 SCIENZA E TECNOLOGIA

Su questo dataset il clustering ottenuto con il Latent Semantic Indexing appare buono: i primi 10-20 cluster sono infatti tutti costituiti da news coerenti fra loro, e con dimensioni confrontabili con i cluster forniti.
Basandoci sui primi cluster di `scienze-tecnologia_0` e su quelli generati con LSI, possiamo notare che:

- Il primo cluster in `scienze-tecnologia_0` è significativo in quanto è un cluster che contiene 164 news che parlano di eventi legati al mondo della tecnologia, come l'evento di Google del 29 Ottobre e l'evento di Apple del 23 Ottobre. Nel clustering generato dal Latent Sematic Indexing, invece, è presente un cluster di 30 news, in ottava posizione, le cui notizie trattano, però, solo argomenti legati ai nuovi Macbook Pro con retina display e i relativi eventi Apple che lo riguardano.

- Il secondo cluster in `scienze-tecnologia_0` è composto da 106 news, ma non non sembra essere molto significativo: gli argomenti trattati dalle news in esso presenti sono, infatti, di natura molto diversa: videogiochi per console/pc, smartphone (soprattutto dell'Asus Padfone) e Windows 8. Nel clustering generato dal Latent Sematic Indexing, invece, troviamo due cluster significativi, entrambi composti da 46 news:
  - Il primo, collocato in quarta posizione, contenente news che parlano di smartphone;

  - Il secondo, collocato in quinta posizione, contenente news che parlano, principalmente, dell'Asus Padfone.

- Il terzo cluster in `scienze-tecnologia_0` conta 70 news ed è significativo, considerando che le notizie in esso presenti trattano argomenti riguardanti la telefonia: l'entrata sul mercato italiano dell'azienda ZTE e l'arrivo dell'LTE in Italia. Nel clustering generato dal Latent Sematic Indexing è presente un cluster molto simile, in sesta posizione, composto da 32 news che trattano, principalmente, dell'arrivo dell'LTE in Italia. Purtroppo, in questo cluster, sono però evidenti, nella parte finale di quest'ultimo, diversi outlier (una decina di news) che parlano di tablet o di aziende di elettronica di consumo.

- Il quarto cluster in `scienze-tecnologia_0` contiene 48 news ed è molto significativo: le news in esso presenti, infatti, parlano della nuova versione di Android Jelly Bean. Nel clustering generato dal Latent Sematic Indexing è presente un cluster molto simile, in seconda posizione, composto da 96 news che parlano sì, della nuova versione di Android Jelly Bean, ma anche del nuovo Samsung Galaxy S3.

- Il quinto cluster in `scienze-tecnologia_0` contiene 42 news che trattano tutte argomenti inerenti ad iPhone ed iPad. Nel clustering generato dal Latent Sematic Indexing è invece presente, in terza posizione, un cluster di 69 news che parlano solo di argomenti interenti ad iPad.

In generale, i cluster di grandi dimensioni presenti in `scienze-tecnologia_0` si possono trovare anche nel clustering generato con il Latent Semantic Indexing, ma con alcune piccole differenze sugli argomenti trattati: i cluster generati con LSI, infatti, contengono news che parlano di uno specifico argomento mentre, quelli presenti in `scienze-tecnologia_0`, se di grandi dimensioni, tendono a contenere news che trattano di argomenti che, concettualmente, potrebbero essere ulteriormente suddivisi per creare dei cluster più specifici e di dimensioni inferiori.
Per i cluster di piccole dimensioni presenti in `scienze-tecnologia_0`, invece, possiamo affermare che:

- Alcuni di essi, come quelli su Xbox Music, Call Of Duty, o sui nuovi smartphone con Windows Phone 8, sono presenti anche nel clustering generato con il Latent Semantic Indexing, ma con dimensioni maggiori;

- Alcuni di essi, come quelli su Fastweb o su Felix Baumgartner, sono presenti anche nel clustering generato con il Latent Semantic Indexing, con dimensioni simili;

- Altri, invece, come quelli su Professor Layton o sull'Antispam, non sono presenti nel clustering generato con il Latent Semantic Indexing.

Sebbene nel clustering del Latent Semantic Indexing siano presenti alcuni cluster che non sembrano comparire nel clustering di `scienze-tecnologia_0`, come quelli su Doom 3, Ilva di Taranto, o sui data center di Google, vale anche il viceversa.
Concludendo, possiamo quindi affermare che, il clustering generato con il Latent Semantic Indexing, è paragonabile al clustering di `scienze-tecnologia_0`: si ottengono dei risultati leggermente migliori per quanto riguarda i cluster di grandi dimensioni, mentre non si può dire lo stesso per i cluster di piccole dimensioni che si possono trovare in posizione superiore alla ventesima, in quanto tendono a non godere sempre di una buona coesione tra le loro news. Questa caratteristica, invece, è sempre presente nei cluster di piccoli dimensioni in `scienze-tecnologia_0`.

3.6 SPETTACOLI

Su questo dataset il clustering ottenuto con il Latent Semantic Indexing appare buono: i primi cluster sono tutti costituiti da news coerenti fra loro, e i cluster hanno dimensioni confrontabili con i cluster forniti.

Confrontando i cluster con quelli forniti, ciò che si nota è che

- Il primo cluster in `spettacoli_0` non sembra molto significativo: contiene notizie di diverso tipo, riguardanti uscite di film, cd musicali e telefilm non collegate fra loro;

- Il secondo cluster è invece costituito da notizie riguardanti Belen, ed è presente anche nel clustering ottenuto tramite LSI: questo secondo cluster è leggermente più grande (75 news rispetto alle 64 del primo); tra le ultime notizie assegnate al cluster (ovvero, quelle che gli appartengono con meno probabilità) compaiono alcune news (non più di 5) che non sono relative all'argomento.

  Un discorso analogo vale per gli altri cluster presenti nelle prime dieci posizioni nel file `spettacoli_0`: sono presenti anche nel clustering con LSI, in posizioni differenti, ma con più o meno le stesse notizie.

- Il quarto topic, riguardante il programma Uomini e Donne, è il secondo nel clustering con LSI: questo cluster contiene il doppio delle notizie di quello in `spettacoli_0` e, a parte l'ultima notizia, le news sono tutte relative a personaggi di questo programma televisivo.

Anche per altri topic, come ad esempio il topic riguardante Kristen Stewart e Robert Pattinson, il clustering con LSI genera cluster con molte più notizie: in questo caso una possibile motivazione è che il cluster nel file fornito contiene solo notizie riguardanti la vita privata dei due attori, mentre il nostro contiene anche notizie che riguardano i film che i due attori hanno girato insieme.

Su questa categoria possiamo quindi concludere che il clustering con Latent Semantic Indexing ottiene risultati confrontabili con quelli forniti e, in alcuni casi, migliori.

3.7 SPORT

Per questo dataset il clustering ottenuto con LSI presenta dei cluster contenenti news molto coese tra loro, almeno per quanto riguarda i primi 10-20 cluster.
Confrontando il clustering ottenuto con quello in `sport_0`, basandoci sui primi dieci cluster, possiamo affermare che:

- I primi quattro cluster presenti in `sport_0` si trovano anche nel clustering generato con LSI tra le prime dieci posizioni e con dimensioni molto simili (tranne in un caso). Per esempio:
  - Il primo cluster di `sport_0`, contenente 382 news che parlano della partita Juventus-Milan, è presente anche nel clustering LSI in quinta posizione con, però, 41 news;
  - Il secondo cluster di `sport_0`, contenente 61 news che parlano della partita Verona-Livorno e degli insulti degli ultrà rivolti verso un giocatore della stessa partita, è presente anche nel clustering LSI ma sotto forma di due cluster differenti:
    - Il primo, in ottava posizione, contenete 28 news che parlano principalmente degli insulti degli ultrà rivolti verso un giocatore della partita Verona-Livorno;
    - Il secondo, in nona posizione, contenente 26 news che parlano della partita Verona-Livorno.
  - Il terzo cluster di `sport_0`, contenente 51 news che parlano del Chelsea e del Manchester United, è presente anche nel clustering LSI in quarta posizione con 53 news;
  - Il quarto cluster di `sport_0`, contenente 51 news che parlano di Fiorenzo Magni, è presente anche nel clustering LSI in seconda posizione con 67 news;
- La maggior parte dei restanti cluster, presenti nelle prime posizioni in `sport_0`, non sono invece presenti nel clustering generato con LSI. Viceversa, ci sono, però, diversi cluster che sono

presenti nel clustering LSI ma non nel clustering di `sport_0`. Per esempio, i tre cluster di:

- Calciomercato Milan;
- Calciomercato Juventus;
- Calciomercato Inter;

non sono presenti in `sport_0`.

Concludendo, i due clustering sono paragonabili in quanto riescono sì, ad individuare alcuni stessi cluster ma, allo stesso tempo, sebbene LSI contenga dei cluster non presenti in `sport_0`, alcuni cluster che si possono trovare in quest'ultimo non vengono individuati.

4 CONCLUSIONI

I cluster generati dalla libreria GibbsLDA++ sono sicuramente paragonabili ai cluster forniti per il confronto, anche su quei dataset considerati più difficili da clusterizzare, come economia e scienza e tecnologia. In alcuni casi, inoltre, i risultati ottenuti possono essere considerati qualitativamente migliori, come accade, per esempio, sul dataset spettacoli.

D'altro canto, però, si possono notare, principalmente, tre caratteristiche negative che caratterizzano i risultati generati da GibbsLDA++:

1. I cluster trovati non sono sempre privi di outlier . Alcune volte, infatti, questi contengono qualche news non coerente con le altre dello stesso cluster.

2. I cluster significativi sono i primi 10-20. Sopra questa soglia i cluster trovati sono, in genere, piccoli e composti da news tra loro scorrelate.

3. Il tempo richiesto per generare un buon clustering, a partire da un generico dataset, è molto alto[1]. Tale tempo è determinato dal numero di iterazioni che la libreria esegue, ma valori piccoli per questo parametro producono clustering approssimativi.

   Per esempio, per generare uno qualunque dei clustering usati nel Capitolo 3 per il confronto, il tempo richiesto è stato, in media, di 25 minuti su un Macbook Pro con Intel Core 2 Duo 2,4 GHz e 4 GB 667 MHz DDR2.

1 L'esecuzione dell'applicazione con i parametri di default richiede dai 15 ai 30 minuti.

5 MANUALE D'USO

In questo ultimo capitolo viene descritto come compilare ed eseguire il programma sviluppato per fare clustering utilizzando la libreria GibbsLDA++.

5.1 COMPILAZIONE DEI SORGENTI

Dopo aver scompattato il file compresso contenente i sorgenti del programma, posizionarsi all'interno della directory prodotta e digitare da terminale i seguenti comandi:

```
make clean
make all
```

Se la compilazione è andata a buon fine l'eseguibile del programma sarà presente nella directory. Per utilizzarlo una prima volta digitare, sempre da terminale, il comando:

```
./clusteringLDA --help
```

Per poter utilizzare il comando da qualsiasi posizione del file system, aggiungerlo alla propria variabile `PATH`.

5.2 USO

L'uso del programma da riga di comando richiede di specificare il percorso del file che contiene il dataset. Come già riportato, è preferibile che il dataset sia già stato stemmato. È inoltre possibili indicare delle opzioni per l'esecuzione.
Per eseguire il comando su un particolare dataset digitare:

```
./clusteringLDA [options] dataset_file
```

Il programma mette a disposizione più opzioni, una relativa ad ogni parametro configurabile per il clustering, oltre a quelle necessarie a visualizzare informazioni del programma. I parametri si dividono in due categorie: quelli che riguardano la libreria e quelli utili alle fasi di pre e post processing. Inoltre è possibile disabilitare alcuni comportamenti standard del programma. Infine, se un'opzione prevede un parametro, questo è obbligatorio.
La lista delle opzioni possibili è la seguente:

- `--help`: mostra le opzioni del programma;

- `-v`: mostra quali sono i valori dei parametri del programma prima di iniziare l'esecuzione;
- `-a alpha`: permette di impostare il valore del parametro alpha della libreria, tale valore è un di tipo reale;
- `-b beta`: permette di impostare il valore del parametro beta della libreria, tale valore è un di tipo reale;
- `-n clusters`: permette di impostare il numero di cluster che si vogliono generare[1];
- `-t terms`: permette di impostare il numero di termini più probabili che si vogliono visualizzare nel file di output, tale parametro non condiziona il risultato ma solo la sua visualizzazione;
- `-m size`: permette di impostare la dimensione minima, ovvero il minimo numero di documenti, che un cluster deve avere per essere presente sul file di output. I cluster sotto tale soglia non vengono riportati nel risultato;
- `-i iter`: permette di impostare il numero di iterazioni che la libreria esegue per produrre il modello finale, ovvero il clustering su cui si applica poi il post-processing;
- `-s step`: permette di impostare il numero di iterazioni dopo le quali la libreria produce un modello intermedio, utile per avere un confronto con il modello finale;
- `-o file`: permette di indicare il nome del file di output che verrà prodotto;
- `-c clust`: permette di indicare il nome di un modello prodotto dalla libreria (di cui verranno poi considerati il file .theta e .twords) su cui eseguire solo la fase di pre e post processing, escludendo l'esecuzione della libreria che risulta essere la fase computazionalmente più costosa, per ottenere il clustering finale[2];
- `-d stringa`: permette di indicare quali filtri disabilitare attraverso la stringa passata come argomento. I seguenti caratteri, se contenuti nella stringa, permettono di disabilitare il relativo filtro:
  - `'.'`: disabilita il filtro sulla punteggiatura
  - `'s'`: disabilita il filtro sulle stopwords

1 I cluster presenti nel file di output prodotto potrebbero essere di meno a causa di eventuali filtri nella fase di post-processing.

2 Serve comunque indicare il dataset perché l'eventuale filtro sui documenti di un topic lo richiede. Inoltre, se il dataset indicato non è quello da cui si è ottenuto il clustering, il risultato che si ottiene è fuorviante.

- `'w'`: disabilita l'uso degli shingles, si considerano le singole parole
- `'i'`: disabilita il filtro che utilizza il valore dell'idf dei termini
- `'m'`: disabilita il filtro sulla dimensione dei cluster, tutti i cluster sono presenti sul file di output prodotto
- `'p'`: disabilita il filtro sui documenti, nessuno documento viene rimosso dal relativo cluster

eventuali altri caratteri presenti nella stringa non vengono considerati

Una considerazione importante riguarda l'output prodotto. Viene infatti prodotto il clustering solo per il modello finale prodotto dalla libreria. Per ottenere il clustering da modelli intermedi prodotti durante l'esecuzione del programma bisogna rieseguirlo con la relativa opzione (-c modello). I modelli intermedi prodotti, vengono salvati all'interno del cartella `temp_clusteringLDA`, che contiene i file temporanei dell'applicazione. Infine, se un filtro viene disabilitato un eventuale modifica del suo parametro viene ignorata.

BIBLIOGRAFIA

[1] *Gibbs Sampling*. 2013. http://en.wikipedia.org/wiki/Gibbs_sampling.

[2] John Hughes. *Latent Dirichlet Allocation v Latent Semantic Indexing*. 2010. http://www.indiciumweb.co.uk.

[3] Xuan-Hieu Phan and Cam-Tu Nguyen. *GibbsLDA++: A C/C++ Implementation of Latent Dirichlet Allocation*. 2007-2008. http://gibbslda.sourceforge.net/.