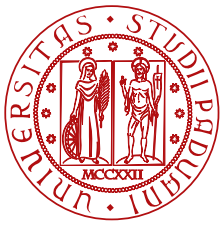

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

Analisi ed applicazione di un algoritmo di Topic Modeling

Relatore
Prof. Di Buccio Emanuele

Laureando
Federico Meneghetti
Matricola N. 2032523

ANNO ACCADEMICO 2023-2024

Data di laurea 27/09/2024

Desidero ringraziare i miei genitori e mio fratello, punti di riferimento per la mia vita e carriera. Ringrazio mia nonna e tutta la mia famiglia, a cui associo i ricordi a cui sono più affezionato. Un pensiero speciale va ai miei amici, per i quali nutro profonda stima. Vorrei inoltre esprimere la mia sincera gratitudine al mio relatore, ai miei compagni universitari e a tutte le persone che mi hanno aiutato e accompagnato durante questo percorso accademico.

Sommario

Questa tesi mira ad approfondire il topic modeling, cioè una tecnica di analisi statistica, utilizzata nell'ambito del text mining e della Knowledge Discovery Database, che permette di ricavare delle informazioni da un insieme molto vasto di dati. Nello specifico, si vogliono determinare i topic, ovvero gli argomenti affrontati nei documenti in esame che vengono modellati come distribuzioni di probabilità sulle parole.

Inizialmente vengono affrontate le applicazioni principali facendo riferimento a vari ambiti tra cui la giurisprudenza e la bioinformatica. Poi, l'elaborato procede ad esaminare nel dettaglio quattro approcci che hanno definito le basi per questa nuova area di ricerca. In particolare, i primi due si contraddistinguono per orientarsi al problema secondo un metodo algebrico lineare. Il primo affrontato è l'Analisi Semantica Latente (LSA), proposta da Deerwester et al. nel 1990, che basandosi sulla decomposizione a valori singolari (SVD) riduce la dimensionalità dei dati testuali per scoprire temi latenti nelle relazioni tra parole e documenti. Di seguito la Non-Negative Matrix Factorization (NMF), definita nel 1999 da Lee e Seung, che rappresenta una estensione diretta di LSA poiché permette di fattorizzare la matrice in ingresso in componenti non negative, offrendo maggiore interpretabilità dei risultati. Il terzo modello è l'Analisi Semantica Latente Probabilistica (PLSA), formulata da Hofmann nel 1999, che introduce un cambio di paradigma sviluppando un processo generativo probabilistico per spiegare la distribuzione dei temi nei documenti. Infine, viene descritta la *Latent Dirichlet Allocation (LDA)*, introdotta nel 2003 da Blei, che rappresenta il modello più rappresentativo poiché evolve il precedente sfruttando la distribuzione di Dirichlet in un approccio bayesiano che si dimostra efficace in diversi contesti applicativi. La discussione dei modelli, oltre a comprendere esempi e spiegazioni di tecniche di ottimizzazione e inferenza, come l'algoritmo *Expectation-Maximization (EM)* e il *Gibbs Sampling*, è integrata dalla trattazione di specifiche metriche dette di *topic coherence* che permettono di valutare i risultati ottenuti dalle analisi dei testi. L'ultimo capitolo include un'applicazione pratica del modello LDA in cui vengono affrontate tutti le fasi del processo: dalla pre-elaborazione del dataset, all'interpretazione degli output. In sintesi, questa tesi ha l'obiettivo di approfondire alcuni approcci rappresentativi di topic modeling tra quelli proposti in letteratura, focalizzandosi su una spiegazione approfondita di tutti i loro aspetti.

Indice

Elenco delle figure

Capitolo 1

Introduzione

Da sempre il sapere è la più affascinante e impegnativa aspirazione della mente umana. Nel corso dei secoli passati la conoscenza si è evoluta e sviluppata attraverso i libri ovvero la carta stampata e solo in epoche relativamente recenti, con l'avvento dei mezzi di comunicazione, l'apprendere ma soprattutto il discernere ha acquisito altri metodi e strumenti. Oggigiorno, uno dei mezzi a nostra disposizione che ci consente di accrescere lo scibile umano molto più velocemente che in passato, è l'informatica. La computazione logica dei dati strutturati mediante l'applicazione di sistemi elettronici e automatizzati ci consente un accesso rapido ed efficiente alle informazioni. I repentini cambiamenti degli ultimi anni, caratterizzati dall'aumento della digitalizzazione, hanno prodotto un numero enorme di dati che, essendo per la maggior parte non strutturati, sono di difficile analisi. Il topic modeling, in tal senso, è una delle più potenti tecniche in quanto ci consente di catalogare automaticamente i testi per argomenti, offrendoci così le informazioni generali dei dati che stiamo osservando al fine di selezionare e approfondire ciò che corrisponde di più al nostro interesse.

Capitolo 2

Definizione e applicazioni del Topic Modelling

2.1 Cos'è il Topic Modelling

Il topic modeling è una tecnica di analisi statistica che permette di ricavare delle informazioni da un insieme molto vasto di dati o documenti con lo scopo di fornire misure quantitative che possono essere utilizzate per identificare il contenuto dei testi, ossia definire gli argomenti trattati, tracciare le variazioni nel tempo ed esprimere le somiglianze tra gli stessi.

2.2 Knowledge Discovery Database e text mining

Si può considerare il topic modeling come parte del più ampio processo Knowledge Discovery in Databases (KDD): ovvero un processo non banale atto all'identificazione e estrazione di pattern e informazioni potenzialmente utili da grandi volumi di dati [1].

Nello specifico, il KDD si suddivide nelle seguenti fasi:

1. Selezione: viene creato un dataset, distinguendo un sottoinsieme dei dati totali disponibili. Si eliminano quindi le parti superflue, con lo scopo di cernire le informazioni più importanti per il problema in esame e migliorare l'efficienza delle fasi successive.

2. Pre-processing: vengono effettuate le operazioni di "pulizia" dei dati, come la correzione di errori, la normalizzazione, e la rimozione degli elementi non rilevanti. L'obiettivo è ottenere un dataset coerente e privo di "rumore" che potrebbe compromettere le analisi seguenti.

3. Trasformazione: i dati pre-eleborati vengono modificati per essere compatibili ad un formato adatto all'applicazione di algoritmi di data mining. In questa fase possono esse-

re utilizzate tecniche di riduzione della dimensionalità o di aggregazione per rendere il dataset più gestibile.

4. Data mining: si tratta della fase fondamentale dell'intero processo in quanto vengono applicati sofisticati strumenti di analisi per estrarre pattern significativi ed informazioni nascoste dai dati. Viene definito text mining un'applicazione specifica del data mining a testi non strutturati. In quest'ambito, il topic modeling è una delle tecniche più efficaci per analizzare grandi raccolte di documenti per individuare in modo automatico gli argomenti (topic) trattati nei testi. Si possono adottare altri algoritmi a seconda del tipo di informazione che si vuole estrarre.

5. Valutazione e interpretazione: vengono valutati i risultati del data mininig allo scopo di stabilirne utilità e validità. In questo stadio si include anche la visualizzazione delle informazioni ottenute per renderle applicabili ad un uso concreto.

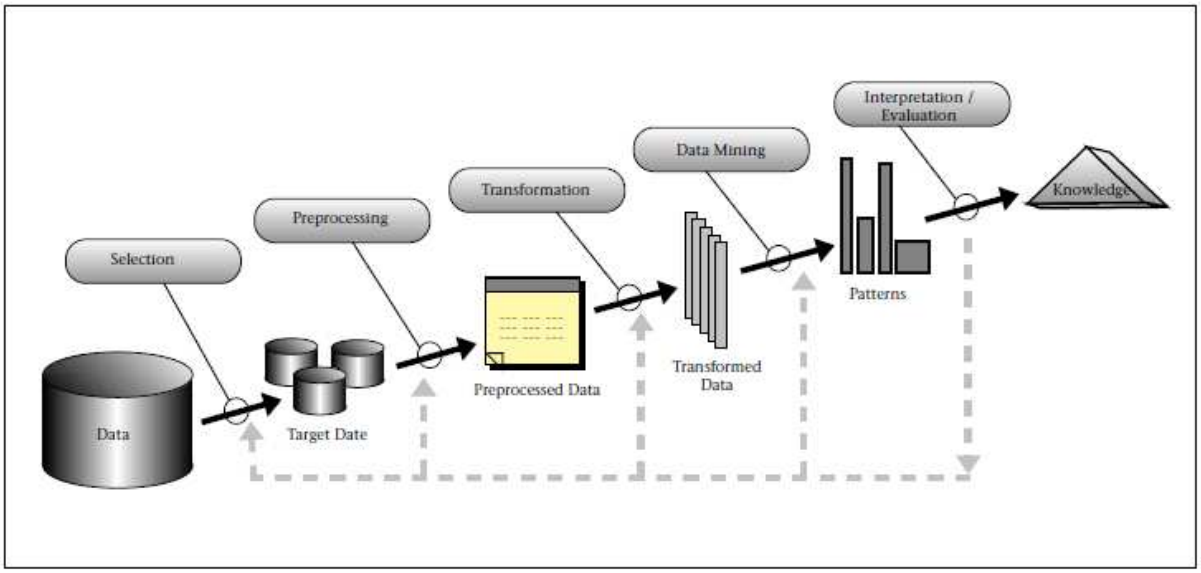

Figura 2.1: Fasi che compongono il processo KDD [1]

2.3 Rapida panoramica di alcune delle applicazioni

I modelli di analisi progettati dai ricercatori, genericamente intesi come topic models (TM), sono dunque strumenti molto sofisticati concepiti per cercare di risolvere il problema della gestione dei dati. La loro applicazione avviene frequentemente in diversi ambiti, come nel campo della letteratura scientifica dove vengono usati non solo per classificare le opere, ma anche per individuare l'attinenza degli argomenti trattati nei vari lavori con l'oggetto degli studi in fase

di elaborazione e ricerca. Nell'industria e nel commercio, consideriamo un'azienda che produce milioni di byte in un giorno, che possono derivare dalle email di partner commerciali, da questionari online o dalle interazioni degli utenti sui canali social. Tutte queste informazioni possono essere valorizzate per ottimizzare non solo prodotti e servizi ma anche per le esperienze stesse dei clienti e quindi fidelizzarli. Ipotizziamo un'azienda che voglia sapere cosa dicono i consumatori di un prodotto sui social network. In questo caso è possibile combinare le tecniche di topic modelling con quelle di analisi del sentiment [1] per identificare la qualità dell'esperienza utente che può essere positiva, negativa o neutra. Facciamo un esempio concreto: si considerino dei commenti che contengono le parole fotocamera, cellulare, qualità, video, lentezza, giochi, inefficienza, app e produttività, che chiaramente hanno come topic principale lo smartphone. I TM, attraverso un metodo statistico, permettono di calcolare le percentuali di riferimento ad argomenti più specifici: il 60% delle opinioni potrebbe riguardare l'efficienza della fotocamera e della buona risoluzione video, il 30% di scarsa applicazione ai giochi e il 10% dell'inefficienza di app legate al lavoro e alla produttività. Dunque, si possono analizzare in modo efficace i feedback per far fronte a problematicità e valorizzare i punti di forza negli sviluppi futuri. Per quanto riguarda i Media e gli articoli giornalistici, applicando i metodi fin qui esposti, si può anche rilevare la frequenza con cui vengono affrontati argomenti differenti, tra cui lo sport, l'intrattenimento, l'attualità, la politica o l'economia; cogliendone la rilevanza e come questi cambino nel corso del tempo, si comprende il cambiamento della nostra società. In antologia e letteratura, in grandi corpora di opere, l'applicazione dei topic models permette di identificare, attraverso l'analisi di termini ricorrenti, i temi nascosti oppure come soggetti quali la guerra, l'amore, la natura ecc.. siano trattati dai vari autori appartenenti ad epoche diverse. In sintesi, le aree di interesse a cui possono essere applicati sono tantissime e tutte molto utili e interessanti; la tesi si soffermerà su due: la giurisprudenza e la bioinformatica, di cui segue un breve approfondimento.

2.3.1 Giurisprudenza

La giustizia italiana ha problemi di produttività [3] dovuti anche tra l'altro al grande numero di cause in sede civile e penale che generano una crescente produzione di documenti, dei quali è necessario innanzitutto identificare i "temi" per poi dotare il corpus di una struttura tematica e facilitare così l'analisi dei casi e il reperimento dei documenti da parte dei professionisti. L'utilità dell'applicazione dei TM è data dalla facilitazione della ricerca dei precedenti nelle sentenze nell'ambito di una determinata fattispecie, ma anche nell'assegnazione dei ricorsi pendenti presso le diverse sezioni della Corte di Cassazione specializzate nelle varie aree di competenza.

[1] Indagine conoscitiva, condotta con metodi di statistica linguistica, sull'opinione, sullo stato d'animo e sulle aspettative degli utenti della rete telematica [2].

Analisi delle sentenze della Corte Costituzionale italiana dal 1956 al 2022 tramite topic modeling Nell'ambito del Programma per la qualità del sistema Giustizia (Laboratorio di AI e Data Science per Giuristi), il lavoro documentato in [4] riporta l'analisi della collezione completa delle sentenze della Corte costituzionale Italiana dal 1956 al 2022 costituita da 21500 atti per osservare come i temi sono cambiati nel tempo e come sono correlati a periodi o eventi storici e culturali noti dell'epoca. In particolare, l'elaborato è basato sull'applicazione del modello Latent Direct Allocation (sezione 6) su un dataset composto da 64.000 parole e pre-elaborato per eliminare tutti i fattori confondenti, ovvero privato delle cosiddette stopwords: abbreviazioni, segni di punteggiatura, cognomi, avverbi, congiunzioni, pronomi e preposizioni molto comuni e di uso regolare.

Il primo passaggio fondamentale consisteva nello stabilire il numero "ottimo" di topic da considerare; per individuare tale valore sono state prese in considerazione metriche di "coherence" (Roder et al [5], 2015) e di "silhouettes" (Rousseeu [6] 1987), calcolate sui vettori *tf-idf* (Sammut e Webb [7], 2010), ottenendo 15 come numero ottimale. Si è potuto quindi ricavare l'evoluzione dei topic ad intervalli di 5 anni con spostamenti di un anno: sono stati conteggiati i documenti appartenenti a ciascuno dei 15 argomenti e, poiché il numero di documenti varia nel tempo, si è osservato come questi aumentino o diminuiscano consentendo analisi comparative tra diverse giurisdizioni e periodi storici.

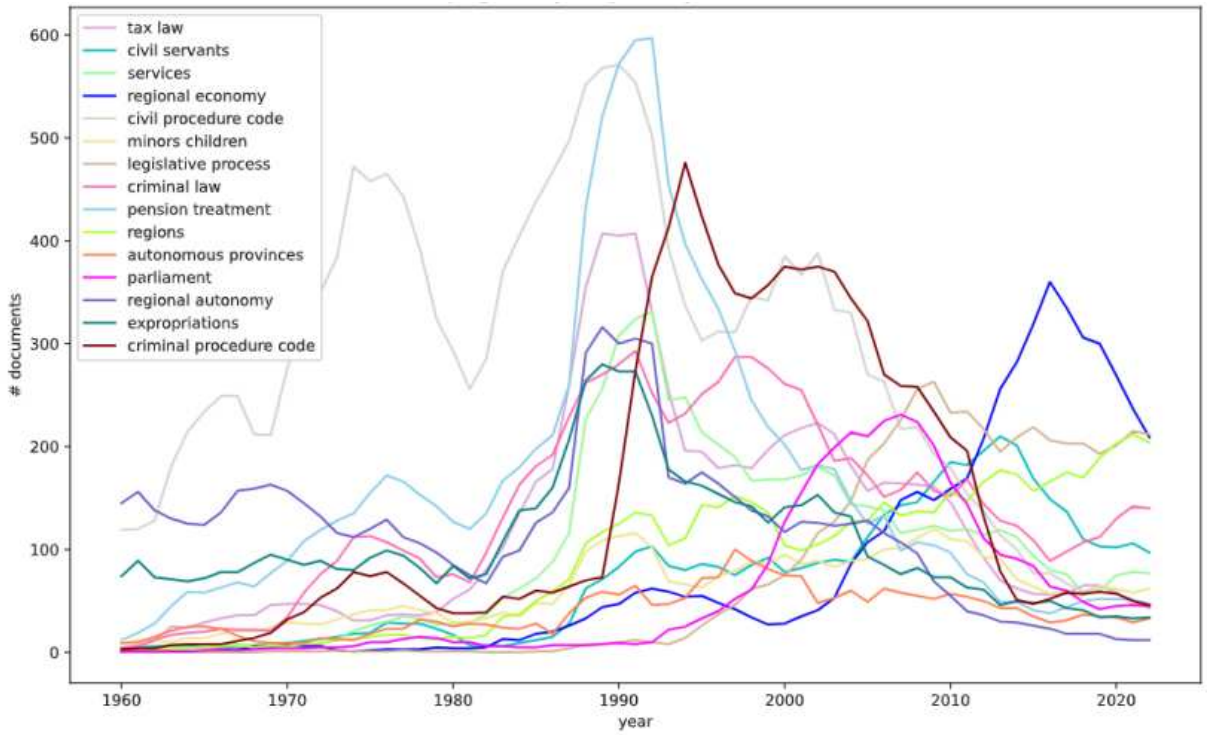

Figura 2.2: Numero di documenti appartenenti a ciascuno dei 15 temi modellati dall'analisi delle sentenze della Corte Costituzionale italiana dal 1956 al 2022 [4]

2.3.2 Bioinformatica

Con il progresso della ricerca scientifica degli ultimi anni abbiamo assistito a una enorme crescita dei dati biologici e di conseguenza la necessità di estrarre conoscenza e relazioni nascoste da questi. I ricercatori hanno quindi iniziato ad applicare i metodi di topic modelling alla bioinformatica concentrandosi principalmente su tre aspetti: analisi dei cluster, classificazione dei dati biologici ed estrazione delle relative caratteristiche [8]. In particolare, il clustering è una tecnica di analisi dei dati atta a selezionare elementi simili tra loro al fine di raggrupparli in insiemi chiamati cluster. L'obiettivo è trovare collezioni che condividono pattern e proprietà comuni. Nel clustering tradizionale, ogni documento può appartenere a un solo cluster, ossia viene assegnato ad una sola categoria. I documenti, tuttavia, possono trattare più argomenti contemporaneamente: una rivista ad esempio potrebbe parlare sia di chimica che di biologia, mentre un campione biologico potrebbe contenere geni appartenenti a più gruppi funzionali. Dunque, il topic modelling da forma ad una distribuzione di probabilità dei documenti sugli argomenti, ed ognuno di essi potrà appartenere a più cluster contemporaneamente, con diversi gradi di affinità. Si noti che questa analisi permette di determinare solo gli argomenti ma non individua le etichette corrispondenti, pertanto sono stati sviluppati dei modelli supervisionati anche per la classificazione dei dati: ovvero si associano gli argomenti scoperti ad etichette biologiche preesistenti (come tipi di cellule o patologie). Infine, l'ultimo obiettivo è quello di estrarre delle caratteristiche dai dati. Di fatto, il topic modelling rappresenta un metodo per la riduzione della dimensionalità dei dati, i quali vengono proiettati in uno spazio di argomenti latenti ridotto. Questo consente di diminuire la complessità e permette di focalizzare gli studi nei pattern più significativi favorendo la diagnosi di malattie o la comprensione dei processi biologici.

Introduciamo un esempio concreto che prevede l'utilizzo di uno strumento molto importante in microbiologia ovvero il microarray di espressione. Si tratta di dispositivi noti anche come chip a DNA o biochip che contengono molecole di DNA dette probes (sonde) attaccate a un supporto solido, generalmente vetro o plastica [9]. Questi dispositivi consentono l'analisi del genoma attraverso la costruzione di mappe di espressione e permettono di studiare migliaia di geni contemporaneamente in diverse condizioni sperimentali. Quindi, una volta ottenuti e normalizzati i dati grezzi, vengono utilizzati algoritmi di clustering per raggruppare i geni in insiemi caratterizzati da espressioni simili, ovvero i topic. Ogni tema identificato viene annotato con informazioni biologiche rilevanti che verranno analizzate per individuare pattern significativi che, a loro volta, saranno sottoposti a validazione. Le applicazioni sono molte, tra cui lo studio dell'espressione genica, l'identificazione di mutazioni geniche e delle alterazioni genomiche. Un esempio è rappresentato dall'uso del topic modeling per identificare i geni delle cellule tumo-

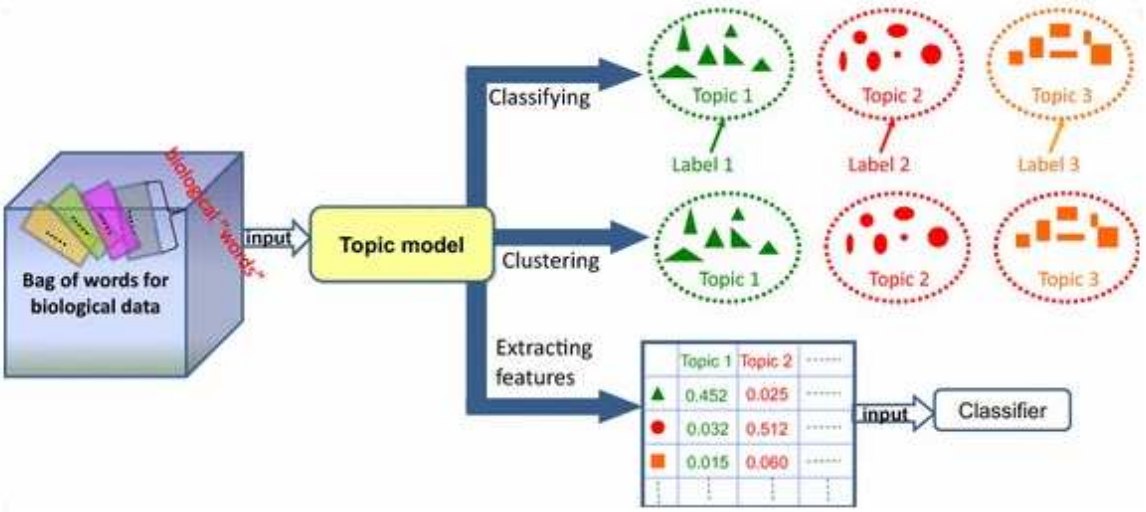

Figura 2.3: Obiettivi del topic modeling in bioinformatica [8]

rali che rispondono a un determinato trattamento farmacologico. Chiaramente per applicare i TM in questo settore piuttosto che nel text mining, è necessaria una rielaborazione dei modelli in cui l'interpretazione degli elementi costitutivi (spiegati nella sezione 3.1) è illustrata della tabella 2.1. In sintesi, l'applicazione dei topic models alla bioinformatica sono solo all'inizio ed i presupposti per il futuro sono molto promettenti nell'ambito di molti campi della ricerca biomedica.

Parole	**Topic**	**Documenti**	**Corpus**
Geni	Gruppi funzionali	Campioni	Dati di microarray di espressione

Tabella 2.1: Elementi costitutivi dei topic models nell'ambito dell'analisi genomica.

Capitolo 3

Latent Semantic Analysis (LSA)

Nel campo del topic modeling nel corso del tempo sono stati sviluppati diversi approcci per analizzare e identificare gli argomenti principali affrontati all'interno di un corpus documenti. La Latent Semantic Analysis (LSA), anche detta Latent Semantic Indexing (LSI), è stata introdotta nel 1990 da Deerwester et al. [10], e rappresenta uno dei primi modelli efficaci sviluppati. Essa si basa sulla decomposizione a valori singolari (SVD) per ridurre la dimensionalità dei dati testuali e scoprire temi latenti nelle relazioni tra parole e documenti. In particolare, l'obiettivo è quello di scomporre una matrice iniziale, documenti-parole, in due sottomatrici distinte: documenti-argomenti e argomenti-parole.

3.1 Elementi costitutivi dei topic models

Al fine di formalizzare la definizione dei diversi modelli vengono prima enunciati gli elementi costituenti, che saranno validi per tutta la trattazione della tesi:

- Una **parola** è l'unità base del modello, ovvero è un elemento appartenente a un vocabolario indicizzato $\{1, \ldots, i, \ldots, V\}$.
- Un **documento** è una sequenza di N parole, indicata come $\mathbf{w_d} = \{w_{d1}, w_{d2}, \ldots, w_{dn}, \ldots, w_{dN_d}\}$, dove w_{dn} rappresenta la n-esima parola del documento. Dunque, una parola $w_{dn} \in \{1, \ldots, i, \ldots V\}$ è uno scalare che assume il valore i se l'n-esima parola del d-esimo documento è l'elemento di indice i del vocabolario della collezione.
- Un **corpus** rappresenta una collezione di M documenti definita come $D = \{\mathbf{w}_1, \ldots, \mathbf{w}_d, \ldots, \mathbf{w}_M\} = \{d_1, \ldots, d_j, \ldots, d_M\}$.

- Un **topic** (argomento) è definito come una distribuzione di probabilità sulle parole del vocabolario: ad ogni termine è attribuita una probabilità che descrive quanto essa sia affine per quel determinato tema. I topic sono indicizzati da $\{1, \ldots, k, \ldots, T\}$ e sono rappresentati dalla variabile z. Ogni parola di un documento w_{dn} è associata ad una variabile z_{dn}, che indica il topic a cui essa appartiene. Nello specifico, $z_{dn} \in \{1, \ldots, k, \ldots, T\}$ è uno scalare che assume il valore k se il topic associato all'n-esima parola del d-esimo documento è il k-esimo topic della collezione.

3.2 Generazione matrice documento-termine

Dati M documenti (corpus) formati da V parole (vocabolario), il primo passo è generare una matrice documenti-parole di dimensione $M \times V$. Ogni voce di tale matrice a livello teorico contiene il conteggio del numero di volte in cui la i-esima parola appare nel documento di indice j. Tuttavia, nella pratica i conteggi non funzionano particolarmente bene in quanto non tengono conto dell' *importanza* di un termine per un testo, in relazione anche agli altri documenti della collezione. Per questo motivo, i modelli LSA sostituiscono i conteggi con un punteggio *tf-idf* [7].

3.2.1 Term Frequency – Inverse Document Frequency (tf-idf)

Possiamo considerare il peso *tf-idf* come composto da due componenti distinte [11]:

1. *Term Frequency (TF)*: rappresenta la frequenza del termine nel documento ed è calcolata come:

$$\text{tf}_{i,j} = \frac{n_{i,j}}{|d_j|}$$

   dove $n_{i,j}$ è il numero di occorrenze del termine w_i nel documento d_j, mentre $|d_j|$ è la lunghezza del j-esimo documento in termini di numero di parole.

2. *Inverse Document Frequency (IDF)*: misura l'importanza del termine nella collezione di documenti ed è calcolata come:

$$\text{idf}_i = \log\left(\frac{|D|}{|\{j : w_i \in d_j\}|}\right)$$

   dove $|D| = M$ è il numero totale di documenti nella collezione e $|\{j : w_i \in d_j\}|$ è il numero di documenti che contengono il termine w_i.

Il prodotto di questi due fattori dà la misura del peso *tf-idf* per il termine w_i nel documento d_j:

$$(\text{tf-idf})_{i,j} = \text{tf}_{i,j} \times \text{idf}_i \tag{3.1}$$

Quindi il punteggio *tf-idf* finale per una parola di un documento varia nel modo che segue:

- è elevato quando il termine è presente molte volte in un numero ristretto di documenti;
- assume un valore meno pronunciato se il termine compare poche volte in un documento o se compare in molti documenti;
- è minimo quando un termine occorre in tutti i documenti.

3.2.2 Bags of words

La rappresentazione dei documenti si basa sull'ipotesi Bag of Words (BoW), ossia che l'ordine delle parole nei testi possa essere trascurato. Si tratta di una tecnica utilizzata di frequente nell'elaborazione del linguaggio naturale (NLP) che si fonda sull'assunzione di *scambiabilità* (Aldous, 1985 [12]): ogni parola di un documento costituisce una variabile casuale che segue la stessa distribuzione, indipendentemente dell'ordine con cui era disposta originariamente nel testo. Nella pratica, il BoW di un corpus è descritto proprio una matrice di frequenza documento-termine. Quindi ogni riga della matrice rappresenta un documento, mentre ogni colonna rappresenta una parola del vocabolario. Le voci della matrice invece specificano la frequenza delle parole nei documenti.

L'assunzione Bags of words permette un'implementazione semplice, tuttavia è caratterizzata da diverse limitazioni legate alla mancanza di contesto e alla perdita del significato sequenziale delle parole. Ad esempio, le due frasi "is this a good day" e "this is a good day" vengono considerate equivalenti durante l'analisi dei dati [13]. Inoltre, la rappresentazione tramite BoW richiede un'attenta pre-elaborazione del testo, altrimenti si avrebbe un modello di dimensioni troppo elevate, con conseguenti scarsità e instabilità numeriche.

Queste limitazioni hanno portato allo sviluppo di tecniche più sofisticate, si citano ad esempio i modelli a word embedding, come word2Vec [14], ed i transformer models che si basano sul deep learning, come BERT [15] .

3.3 Decomposizione a valori singolari (SVD)

Ora, lo scopo è fattorizzare la matrice iniziale in sotto-matrici costituenti. Tale concetto è il medesimo di quando scomponiamo un numero nei sui fattori che, se moltiplicati assieme, ci

forniscono il numero originale; come ad esempio $30 = 2 \times 3 \times 5$.

Nella matematica delle matrici, una scomposizione simile è possibile tramite la diagonalizzazione, ossia un tipo particolare di fattorizzazione che si applica alle matrici quadrate. Se una matrice A è diagonalizzabile, allora possiamo scriverla come il prodotto:

$$A = P\Lambda P^{-1}$$

Dove:

- P è una matrice che ha come colonne gli autovettori di A.
- Λ è una matrice che contiene gli autovalori di A disposti lungo la diagonale.
- P^{-1} è l'inversa della matrice P.

La decomposizione a valori singolari (SVD) generalizza questo concetto rendendolo applicabile alle matrici rettangolari e in generale a tutte le matrici non diagonalizzabili.

Data una matrice A di dimensione $M \times V$, dove M rappresenta il numero di documenti e V il numero di termini, la decomposizione SVD scompone **A** in tre matrici [16] [17]:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$$

Qui:

- **U** è una matrice ortogonale di dimensione $M \times M$, le cui colonne $u_1, \ldots, u_M$ sono detti *vettori singolari sinistri* di **A**. In particolare, nel contesto del topic modeling le colonne di **U** rappresentano i documenti espressi come combinazioni lineari degli argomenti latenti del corpus.
- Σ è una matrice diagonale di dimensione $M \times V$ che contiene i *valori singolari* di **A**, ordinati in modo decrescente. Si ha infatti $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$, dove r è il rango della matrice A. I valori singolari σ_i nella matrice rappresentano l'importanza di ciascun argomento latente nei dati.
- $\mathbf{V}^T$ è la trasposta di una matrice ortogonale **V** di dimensione $V \times V$. Le colonne $v_1, \ldots, v_V$ di **V**, ossia le righe di $\mathbf{V}^T$, sono detti *vettori singolari destri* di **A** e rappresentano i termini (o parole) espressi come combinazioni lineari di argomenti latenti.

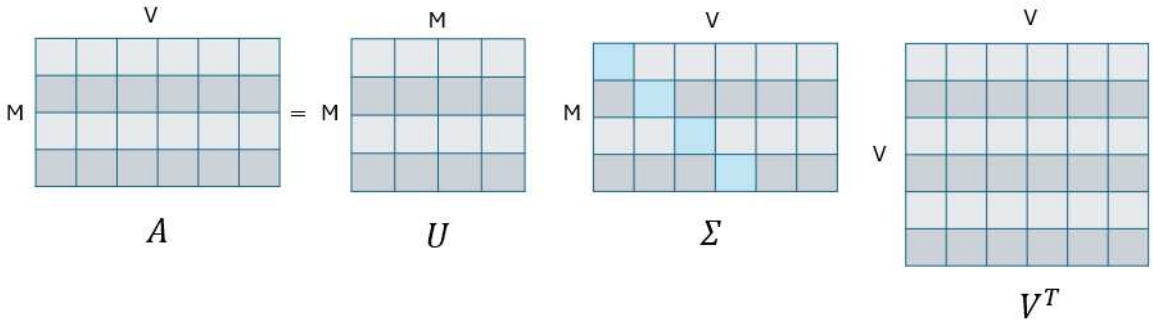

Figura 3.1: Decomposizione a valori singolari.

Valori singolari Matematicamente, i valori singolari σ_i di A sono le radici quadrate degli autovalori non negativi comuni di AA^T e A^TA [1]. Pertanto, se λ_i è un autovalore di AA^T o A^TA, allora $\sigma_i = \sqrt{\lambda_i}$.

Vettori singolari I *vettori singolari sinistri* u_i e *vettori singolari destri* v_i sono rispettivamente gli autovettori di AA^T e A^TA. Valgono quindi le seguenti espressioni:

- $AA^Tu_i = \lambda_i u_i$ dove λ_i è l'autovalore associato a u_i e $\sigma_i = \sqrt{\lambda_i}$.
- $A^TAv_i = \lambda_i v_i$ dove λ_i è l'autovalore associato a v_i e $\sigma_i = \sqrt{\lambda_i}$.

Si noti che i vettori singolari u_i e v_i, a differenza dei valori singolari σ_i, non sono univocamente determinati.

3.3.1 SVD troncato e riduzione della dimensionalità

Se la matrice A documento-termine è di grande dimensioni, la decomposizione SVD può risultare troppo complessa ed includere anche elementi che non sono essenziali, detti "rumore". Idealmente, si vuole un numero sufficiente di dimensioni per catturare la struttura latente della matrice documento-termine, ma non troppe perché altrimenti si considererebbero temi non rilevanti per la collezione in esame. Ad esempio, per descrivere un insieme di documenti vorremmo utilizzare una decina di argomenti principali e non svariate centinaia. Dunque, al fine di semplificare l'analisi e migliorare l'interpretazione dei topic trattati nei documenti, si effettua una versione troncata della SVD, che seleziona solo i primi T valori singolari più grandi.[2] Si può quindi approssimare la matrice in ingresso A nel seguente modo [10] [18] [19] :

[1] Le matrici AA^T e A^TA hanno gli stessi autovalori, tuttavia può variare al il numero degli autovalori nulli.

[2] La scelta del numero T appropriato di dimensioni è una questione di ricerca aperta. In questa tesi viene approfondito un possibile approccio nella sezione 7.2

$$\mathbf{A} \approx \mathbf{U}_t \Sigma_t \mathbf{V}_t^T$$

Dove:

- $\mathbf{U}_t$ ora è una matrice di dimensione $M \times T$, le cui righe, come detto precentemente, sono ”vettori documento” che descrivono quanto un documento è associato a ciascun argomento.

- Σ_t è una matrice diagonale di dimensione $T \times T$, contenente solo i primi T valori singolari.

- $\mathbf{V}_t^T$ è una matrice di dimensione $T \times V$, le cui righe sono ”vettori argomenti” che descrivono quanto un topic è associato a ciascuna parola.

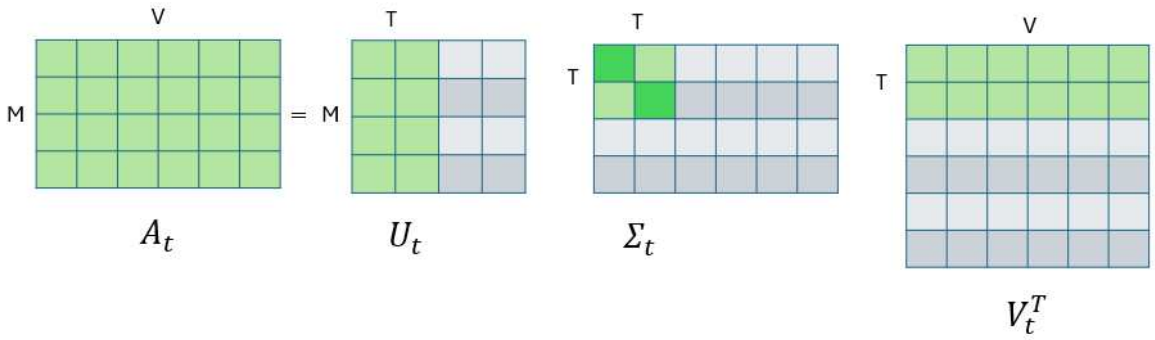

Figura 3.2: Decomposizione a valori singolari troncata.

Due matrici rettangolari $A \in \mathbb{R}^{p \times t}$ e $B \in \mathbb{R}^{t \times r}$ si dicono compatibili se condividono la dimensione intermedia t e, per esse, è definito il prodotto righe per colonne, $C = A \times B$. Se si volesse moltiplicare una serie $\{A_1, A_2, \ldots A_n\}$ di n matrici rettangolari è necessario che esse siano compatibili: ovvero la dimensione di colonna della matrice A_i sia uguale alla dimensione di riga della matrice successiva A_{i+1}, per $1 \leq i < n$. Nel caso della SVD troncata, si definisce quindi una dimensione comune $t = T$ rappresentante del numero di topic che permette di realizzare il prodotto tra la catena di matrici approssimando il risultato.

3.4 Interpretazione ed esempio

doc	Titolo (testo)
d1	Human machine interface for ABC computer applications
d2	A survey of user opinion of computer system response time
d3	The EPS user interface management system
d4	System and human system engineering testing of EPS
d5	Relation of user perceived response time to error measurement
d6	The generation of random, binary, ordered trees
d7	The intersection graph of paths in trees
d8	Graph minors IV: Widths of trees and well-quasi-ordering
d9	Graph minors: A survey

Per analizzare un corpus di 9 documenti [3] il modello prevede la formazione della matrice A, costituita dal conteggio dei termini nei rispettivi testi (non vengono incluse le stopwords), su cui viene applicata la decomposizione SVD per ottenere le sottomatrici costituenti.

	human	interface	computer	user	system	response	time	EPS	survey	trees	graph	minors
d1	1	1	1	0	0	0	0	0	0	0	0	0
d2	0	0	1	1	1	1	1	0	1	0	0	0
d3	0	1	0	1	1	0	0	1	0	1	0	0
d4	1	0	0	0	2	0	0	1	0	1	0	0
d5	0	0	0	1	0	1	1	0	0	0	0	0
d6	0	0	0	0	0	0	0	0	0	1	0	0
d7	0	0	0	0	0	0	0	0	0	1	1	0
d8	0	0	0	0	0	0	0	0	0	1	1	1
d9	0	0	0	0	0	0	0	0	1	0	1	1

$$U = \begin{bmatrix} -0.20 & -0.06 & 0.11 & -0.95 & 0.05 & -0.08 & -0.18 & 0.01 & 0.06 \\ -0.61 & 0.17 & -0.50 & -0.03 & -0.21 & -0.26 & 0.43 & -0.05 & -0.24 \\ -0.46 & -0.13 & 0.21 & 0.04 & 0.38 & 0.72 & 0.24 & -0.01 & -0.02 \\ -0.54 & -0.23 & 0.57 & 0.27 & -0.21 & -0.37 & -0.26 & 0.02 & 0.08 \\ -0.28 & 0.11 & -0.51 & 0.15 & 0.33 & 0.03 & -0.67 & 0.06 & 0.26 \\ -0.00 & 0.19 & 0.10 & 0.02 & 0.39 & -0.30 & 0.34 & -0.45 & 0.62 \\ -0.01 & 0.44 & 0.19 & 0.02 & 0.35 & -0.21 & 0.15 & 0.76 & -0.02 \\ -0.02 & 0.62 & 0.25 & 0.01 & 0.15 & 0.00 & -0.25 & -0.45 & -0.52 \\ -0.08 & 0.53 & 0.08 & -0.02 & -0.60 & 0.36 & -0.04 & 0.07 & 0.45 \end{bmatrix}$$

[3] L'esempio è stato sviluppato sulla base di quello presente in [10]

$$\Sigma = \begin{bmatrix} 3.34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.54 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.35 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.64 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.50 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.31 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.85 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.56 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.36 \end{bmatrix}$$

$$V^T = \begin{bmatrix} -0.22 & -0.20 & -0.24 & -0.40 & -0.64 & -0.27 & -0.27 & -0.30 & -0.21 & -0.01 & -0.04 & -0.03 \\ -0.11 & -0.07 & 0.04 & 0.06 & -0.17 & 0.11 & 0.11 & -0.14 & 0.27 & 0.49 & 0.62 & 0.45 \\ 0.29 & 0.14 & -0.16 & -0.34 & 0.36 & -0.43 & -0.43 & 0.33 & -0.18 & 0.23 & 0.22 & 0.14 \\ -0.41 & -0.55 & -0.59 & 0.10 & 0.33 & 0.07 & 0.07 & 0.19 & -0.03 & 0.02 & 0.00 & -0.01 \\ -0.11 & 0.28 & -0.11 & 0.33 & -0.16 & 0.08 & 0.08 & 0.11 & -0.54 & 0.59 & -0.07 & -0.30 \\ -0.34 & 0.50 & -0.25 & 0.38 & -0.21 & -0.17 & -0.17 & 0.27 & 0.08 & -0.39 & 0.11 & 0.28 \\ -0.52 & 0.07 & 0.30 & -0.00 & 0.17 & -0.28 & -0.28 & -0.03 & 0.47 & 0.29 & -0.16 & -0.34 \\ 0.06 & 0.01 & -0.06 & 0.00 & -0.03 & 0.02 & 0.02 & 0.02 & 0.04 & -0.25 & 0.68 & -0.68 \\ 0.41 & 0.11 & -0.49 & -0.01 & -0.27 & 0.05 & 0.05 & 0.17 & 0.58 & 0.23 & -0.23 & -0.18 \end{bmatrix}$$

Applicando l'SVD troncato si ottiene:

$$A \approx \begin{bmatrix} 0.24 & 0.18 & 0.11 & 0.17 & 0.54 & 0.05 & 0.05 & 0.30 & 0.05 & -0.00 & -0.01 & -0.01 \\ 0.06 & 0.21 & 0.70 & 1.24 & 0.81 & 1.08 & 1.08 & 0.16 & 0.74 & -0.04 & 0.07 & 0.09 \\ 0.52 & 0.39 & 0.28 & 0.44 & 1.23 & 0.17 & 0.17 & 0.67 & 0.14 & -0.03 & -0.04 & -0.03 \\ 0.86 & 0.58 & 0.19 & 0.24 & 1.75 & -0.15 & -0.15 & 1.07 & -0.03 & 0.04 & -0.00 & -0.02 \\ -0.17 & 0.00 & 0.43 & 0.79 & 0.13 & 0.78 & 0.78 & -0.15 & 0.48 & -0.13 & -0.06 & -0.02 \\ 0.01 & -0.00 & -0.01 & -0.04 & 0.01 & -0.04 & -0.04 & 0.01 & 0.10 & 0.29 & 0.36 & 0.25 \\ 0.02 & -0.01 & -0.01 & -0.07 & 0.01 & -0.06 & -0.06 & 0.01 & 0.23 & 0.65 & 0.80 & 0.57 \\ 0.01 & -0.02 & -0.01 & -0.08 & 0.01 & -0.06 & -0.06 & 0.00 & 0.34 & 0.90 & 1.11 & 0.79 \\ -0.04 & -0.02 & 0.09 & 0.12 & 0.02 & 0.14 & 0.14 & -0.05 & 0.39 & 0.71 & 0.89 & 0.64 \end{bmatrix}$$

Come si nota, per garantire l'ortogonalità di U e V è possibile che alcuni elementi di queste matrici siano negativi. Questo chiaramente causa alcuni problemi di interpretazione poiché quando si ha un coefficiente minore di zero, esso comporta che, in una specifica combinazione lineare, viene sottratto un contributo invece che venire aggiunto. Per evitare tale problematica, è stata introdotta la Non-Negative Matrix Factorization che fornisce una rappresentazione basata

su parti dove le componenti sono sempre positive e più sparse, ovvero con un elevato numero di elementi nulli. In ogni caso, analizzando la matrice U_t è possibile vedere le associazioni dei documenti con gli argomenti latenti. Ad esempio, il documento d_4 assume un valore più elevato in relazione al terzo topic, indicando una maggiore correlazione con quest'ultimo. Il terzo argomento, a sua volta, è il meno rilevante dei tre nel corpus (matrice Σ_t), ed ha come parole più importanti *system*, *EPS* e *human* (matrice V_t^T). Possiamo interpretare quindi questo topic come un sistema di user interface. L'esempio mostrato è da intendersi come una visione semplificata che però permette di esplicare in modo intuitivo l'idea di base su cui si fonda il modello.

$U_t =$

	Topic 1	Topic 2	Topic 3
d1	-0.20	-0.06	0.11
d2	-0.61	0.17	-0.50
d3	-0.46	-0.13	0.21
d4	-0.54	-0.23	0.57
d5	-0.28	0.11	-0.51
d6	-0.00	0.19	0.10
d7	-0.01	0.44	0.19
d8	-0.02	0.62	0.25
d9	-0.08	0.53	0.08

$\Sigma_t =$

Topic 1	Topic 2	Topic 3
3.34	0	0
0	2.54	0
0	0	2.35

$V_t^T =$

	human	interface	computer	user	system	response	time	EPS	survey	trees	graph	minors
Topic 1	-0.22	-0.20	-0.24	-0.40	-0.64	-0.27	-0.27	-0.30	-0.21	-0.01	-0.04	-0.03
Topic 2	-0.11	-0.07	0.04	0.06	-0.17	0.11	0.11	-0.14	0.27	0.49	0.62	0.45
Topic 3	0.29	0.14	-0.16	-0.34	0.36	-0.43	-0.43	0.33	-0.18	0.23	0.22	0.14

Capitolo 4

Non-Negative Matrix Factorization (NMF)

La Non-Negative Matrix Factorization (Lee e Seung [20], 1999) , come la Latent Semantic Analysis (LSA), è un metodo di riduzione della dimensionalità applicato alla matrice documento-termine, che si distingue per la sua capacità di produrre rappresentazioni più interpretabili. A differenza della SVD che genera anche componenti negative, la NMF decompone la matrice principale in due matrici non negative e questo rende tale tecnica particolarmente adatta per contesti in cui i dati, come la frequenza dei termini, sono naturalmente maggiori di zero, facilitando la comprensione dei risultati.

4.1 Generazione matrice documento-termine e decomposizione NMF

Il primo passo, come per LSA, è produrre una matrice documento-termine A di dimensioni $M \times V$, dove M è il numero di documenti, V è il numero di termini ed ogni elemento di A rappresenta il peso *tf-idf* (o il conteggio) dei termini nei rispettivi documenti.

La NMF fattorizza in modo non esatto la matrice A in due matrici non negative [18]:

$$A \approx WH$$

Dove:

- W è una matrice $M \times T$, chiamata matrice documento-argomento. Ogni riga di W rappresenta la distribuzione degli argomenti nei documenti del corpus.
- H è una matrice $T \times V$, detta matrice termine-argomento. Ogni colonna di H descrive quanto un termine è associato a ciascun argomento. Ogni riga definisce la distribuzione degli argomenti nelle parole del corpus.

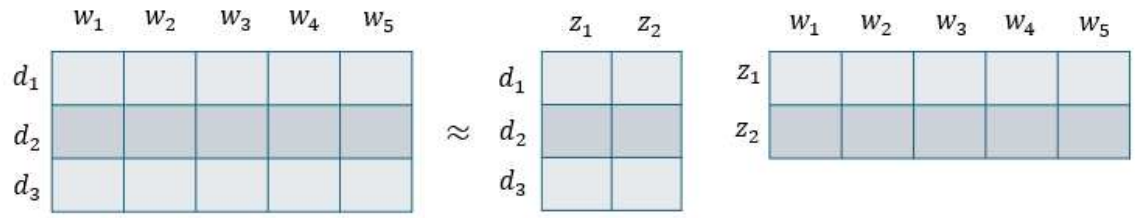

Figura 4.1: Non-Negative Matrix Factorization

Si ricorda che due matrici rettangolari $W \in \mathbb{R}^{M \times T}$ e $H \in \mathbb{R}^{T \times V}$ si dicono compatibili se condividono la dimensione intermedia T ed è definito il loro prodotto righe per colonne, $C = W \times H$. Per quanto riguarda la NMF, si ha che $A \approx C = W \times H$ con la dimensione T (di colonna per W e di riga per H) che rappresenta il numero di topic latenti.

4.2 Problema di Ottimizzazione

La decomposizione NMF può essere formulata come un problema di ottimizzazione in cui l'obiettivo è trovare due matrici non negative W e H che minimizzano l'errore di approssimazione tra il loro prodotto e una matrice in ingresso A. La formulazione matematica è la seguente [21]:

$$\begin{cases} \min \|A - WH\|_F^2 \\ W_{ij} \geq 0, \quad H_{ij} \geq 0, \quad \forall i, j \\ W \in \mathbb{R}^{M \times T}, H \in \mathbb{R}^{T \times V} \end{cases} \tag{4.1}$$

La funzione obiettivo utilizza la norma di Frobenius che, per una generica matrice B, è definita come la radice quadrata della somma dei quadrati dei suoi elementi, e permette di quantificare quanto la matrice sia "lontana" dall'essere nulla. Formalmente, è definita come:

$$\|B\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} B_{ij}^2} \tag{4.2}$$

Nel contesto del NMF, la funzione obiettivo prevede la norma di Frobenius per misurare l'errore di ricostruzione, ossia una misura che quantifica quanto A si avvicina a WH. Di fatto, essa corrisponde alla distanza euclidea tra gli elementi di A e quelli appartenenti al prodotto WH.

$$\|A - WH\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} (A_{ij} - (WH)_{ij})^2} \tag{4.3}$$

Un'altra modalità utilizzata per ottimizzare la differenza tra A e la sua approssimazione WH, è minimizzare la divergenza di Kullback-Leibler definita dall'equazione 4.4.

$$D(A \parallel WH) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(A_{ij} \times \log \left(\frac{A_{ij}}{(WH)_{ij}} + 1 \right) - A_{ij} + (WH)_{ij} \right) \tag{4.4}$$

In questo modo il problema di ottimizzazione diventa:

$$\begin{cases} \min\ D(A \parallel WH) \\ W_{ij} \geq 0, \quad H_{ij} \geq 0, \quad \forall i, j \\ W \in \mathbb{R}^{M \times T}, H \in \mathbb{R}^{T \times V} \end{cases} \tag{4.5}$$

Nonostante la norma di Frobenius sia la più comune per la sua semplicità, essa assume che le differenze tra i dati reali e quelli stimati seguano una distribuzione gaussiana. Tale presupposto tuttavia non è considerabile sempre realistico e, per questo motivo, per dati sparsi basati su conteggi viene spesso minimizzata la divergenza di Kullback-Leibler. In questo caso infatti si assume che gli errori seguano una distribuzione di Poisson, che è più adatta per le ipotesi in cui ci sono molti zeri. Dunque, a seconda degli input può variare la scelta per avere una funzione obiettivo migliore.

4.3 Algoritmi risolutivi

Esistono vari algoritmi per ottimizzare W e H con scopo di approssimare al meglio la matrice originale A. In generale, essi si contraddistinguono per essere iterativi e caratterizzati dalle seguenti fasi: inizializzazione, ottimizzazione e arresto.

4.3.1 Inizializzazione

La fase di inizializzazione consiste nel fornire valori iniziali alle matrici W e H. Si noti che il problema di ottimizzazione non è convesso, quindi sussiste la possibilità che gli algoritmi possano convergere e rimanere bloccati in punti di minimo locale.

Nello specifico, un problema di ottimizzazione del tipo "*min* $f(x)$ soggetto a $x \in X \subseteq R^n$" si dice di programmazione convessa se X è convesso e $f(x)$ è una funzione convessa su X [22]. Se fossero soddisfatte queste ipotesi, non ci sarebbero complicazioni poiché ogni punto di minimo locale è anche un minimo globale; in caso contrario invece a partire da diverse inizializzazioni le iterazioni potrebbero convergere in punti differenti, non garantendo sempre la soluzione ottima globale.

Ad ora, non ci sono giustificazioni teoriche che garantiscono la bontà delle soluzioni rispetto scelte iniziali, dunque la decisione di quali strategie adottare viene presa su studi preliminari, che comprendono fattori come: le caratteristiche del problema, la complessità computazionale e le proprietà specifiche dei dati. Di seguito vengono elencate le strategie che sono comunemente utilizzate [23]:

- *Scelta di un sottoinsieme delle colonne di A*: in questo approccio le matrici vengono inizializzate a partire da un sottoinsieme delle colonne della matrice in input. Nello specifico si può porre:

  $$W = A(:, \Omega)$$

  dove Ω è un sottoinsieme di colonne di A con cardinalità T, che è pari al rango di approssimazione desiderato che, nel nostro contesto, sarà il numero di argomenti.

- *Random*: si tratta del modo più semplice con cui vengono generati gli elementi delle matrici, ovvero in modo casuale all'interno dell'intervallo $[0,1]$. Questo approccio è chiaramente di facile implementazione e non richiede calcoli complessi, tuttavia può risultare meno efficace in termini di convergenza rispetto a strategie più complesse.

- *Tecniche di Clustering*: vengono utilizzate tecniche di clustering come le k-medie o le k-medie sferiche: a partire dai centroidi dei cluster viene generata la matrice W, dove il numero di cluster viene scelto uguale al rango di fattorizzazione T, che corrisponde al numero di argomenti. La matrice H invece rappresenta la matrice di partizionamento, dove $H_{i,j} \neq 0 \Leftrightarrow$ se la voce x_j appartiene al i-esimo cluster. In sintesi, il metodo sfrutta la struttura dei dati per una inizializzazione più sofisticata.

4.3.2 Ottimizzazione

In questa fase, l'obiettivo è migliorare iterativamente le matrici W e H per ridurre l'errore di ricostruzione. Esistono molte classi di algoritmi per la decomposizione NMF, tra cui si citano due categorie di frequente utilizzo:

- *Algoritmi a schema alternato*: in questa tipologia, le matrici W e H vengono ottimizzate alternativamente; ovvero, durante ogni iterazione, si fissa una delle due matrici e viene ottimizzata l'altra. Quindi questo schema semplifica il problema di ottimizzazione in sottoproblemi più semplici che possono essere risolti con diversi approcci, come il Multiplicative Update (MU) [20], l'Alternating Least Squares (ALS) [24] e l'Alternating Non-negative Least Squares (ANLS) [25].

- *Algoritmi con penalizzazione*: vengono introdotte penalizzazioni aggiuntive nella funzione obiettivo al fine di incoraggiare delle proprietà desiderate nelle soluzioni, come la sparsità o delle particolari strutture nelle matrici W e H. Alcuni algoritmi appartenenti a questa classe sono l'Alternating Constrained Least Squares (ACLS) e l'Alternating Hierarchical Constrained Least Squares (AHCLS).

Come esempio viene approfondito brevemente l'algoritmo Multiplicative Update (MU) (Lee e Seung, [20] 1999) che, tramite regole di moltiplicazione delle matrici, permette di minimizzare in progressivamente la funzione obiettivo.

L'approccio generale può essere descritto come segue [21]:

1. Vengono generate le matrici iniziali $W^{(0)} \geq 0$ e $H^{(0)} \geq 0$.
2. Per $t = 1, 2, 3, \ldots$ fare:
   (a) $W^{(t)} = \text{update}(X, H^{(t-1)}, W^{(t-1)})$
   (b) $H^{(t)} = \text{update}(X, W^{(t)}, H^{(t-1)})$
   (c) Se $[D(X; W^{(t-1)}, H^{(t-1)}) - D(X; W^{(t)}, H^{(t)})] \leq \epsilon$ allora fermarsi

dove ϵ è un parametro che definisce una soglia di convergenza come descritto dall'equazione 4.13. Nello specifico, possiamo distinguere due varianti: la prima come funzione obiettivo minimizza la norma di Frobenius, la seconda invece la divergenza di Kullback-Leibler.

NMF: Moltiplicative Update - Norma di Frobenius

1. Vengono inizializzate le matrici iniziali $W^{(0)} \geq 0$ e $H^{(0)} \geq 0$.
2. Per t[1] $= 1, 2, 3, \ldots$ fare:
   (a) Viene massimizzata la matrice W:

$$W^{(t)} \leftarrow W^{(t-1)} \times \frac{AH^{(t-1)^T}}{W^{(t-1)}H^{(t-1)}H^{(t-1)^T}} \tag{4.6}$$

   (b) Viene massimizzata la matrice H:

$$H^{(t)} \leftarrow H^{(t-1)} \times \frac{W^{(t)^T}A}{W^{(t)^T}W^{(t)}H^{(t-1)}} \tag{4.7}$$

[1] La lettera t idica il numero dell'iterazione corrente.

(c) Si ripetono i punti a e b fino a quando non non viene soddisfatto un criterio di convergenza (sezione 4.3.3).

NMF: Aggiornamento Moltiplicativo - Divergenza KL

1. Vengono inizializzate le matrici iniziali $W^{(0)} \geq 0$ e $H^{(0)} \geq 0$.

2. Per $t = 1, 2, 3, \ldots$ fare:

   (a) Viene massimizzata la matrice W:

$$W^{(t)} \leftarrow W^{(t-1)} \times \frac{\frac{A}{W^{(t-1)}H^{(t-1)}} {H^{(t-1)}}^T}{\mathbf{1}{H^{(t-1)}}^T} \tag{4.8}$$

   (b) Viene massimizzata la matrice H:

$$H^{(t)} \leftarrow H^{(t-1)} \times \frac{{W^{(t)}}^T \frac{A}{W^{(t)}H^{(t-1)}}}{{W^{(t)}}^T\mathbf{1}} \tag{4.9}$$

   (c) Si ripetono i punti a e b fino a quando non non viene soddisfatto un criterio di convergenza (sezione 4.3.3).

4.3.3 Arresto

La scelta di una strategia di arresto adeguata è fondamentale per garantire che l'algoritmo converga a una soluzione di alta qualità senza sprecare eccessivo tempo computazionale. Si utilizzano vari criteri per decidere quando terminare l'esecuzione, che per lo più sono basati sull'andamento dell'errore di ricostruzione e sulla variazione progressiva delle soluzioni. Seguono gli approcci più comuni [23]:

- *Residuo relativo*:

$$\frac{\|A - WH\|_F}{\|A\|_F} < \epsilon \tag{4.10}$$

  Viene confrontato l'errore di ricostruzione e con la grandezza della matrice iniziale A. Nel caso in cui tale rapporto sia inferiore ad una soglia di convergenza ϵ l'algoritmo si arresta.

- *Residuo scalato*:

$$\frac{\|A - WH\|_F}{mv} < \epsilon \tag{4.11}$$

Le variabili m e v rappresentano rispettivamente è la dimensione di colonna di A (il numero di documenti M) e il dimensione di riga di A (numero di termini V). Questo approccio è simile al precedente, ma l'errore di ricostruzione viene normalizzato rispetto alla dimensione della matrice A. Come prima il rapporto viene confrontato con valore di soglia ϵ per stabilire quando fermarsi.

- *Confronto con la migliore approssimazione di rango T (SVD troncata)* :

$$\frac{\|A - WH\|_F^2 - \rho}{\rho} < \epsilon \quad \rho = \|A - U_t \Sigma_t V_t^T\|_F \tag{4.12}$$

Questo criterio misura in termini relativi la differenza tra l'errore di ricostruzione e l'errore prodotto dalla migliore approssimazione di rango T, ottenuta tramite decomposizione SVD troncata. Se il risultato del rapporto è inferiore alla soglia ϵ, il processo può arrestarsi poiché ha raggiunto una soluzione prossima a quella che si avrebbe con una decomposizione SVD.

- *Variazione della soluzione*:

$$\max\left(\|H^{(k)} - H^{(k-1)}\|_F, \|W^{(k)} - W^{(k-1)}\|_F\right) < \epsilon \tag{4.13}$$

Questa tecnica prevede di confrontare le soluzioni tra due iterazioni successive. Nel caso in cui le modifiche diventino irrisorie, ossia minori di un numero ϵ, l'algoritmo viene considerato convergente e quindi termina: altre iterazioni infatti produrrebbero modifiche trascurabili.

- *Massimo numero di iterazioni o limite di tempo*: in questo approccio semplicemente viene definito un numero massimo di iterazioni o un limite di tempo di esecuzione massimo, dopo le quali l'algoritmo si ferma.

4.4 Esempio

Considerando come matrice A in input quella realizzata nella sezione 3.4, se si esegue la NMF si ottiene:

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 0.24 & 0.17 & 0.10 & 0.17 & 0.52 & 0.08 & 0.08 & 0.30 & 0.05 & 0.00 & 0.00 & 0.00 \\ 0.18 & 0.15 & 0.69 & 1.20 & 0.72 & 1.12 & 1.12 & 0.22 & 0.71 & 0.02 & 0.03 & 0.05 \\ 0.55 & 0.39 & 0.22 & 0.39 & 1.20 & 0.19 & 0.19 & 0.71 & 0.12 & 0.01 & 0.00 & 0.00 \\ 0.86 & 0.60 & 0.18 & 0.31 & 1.78 & 0.00 & 0.00 & 1.10 & 0.00 & 0.02 & 0.00 & 0.00 \\ 0.00 & 0.02 & 0.47 & 0.82 & 0.26 & 0.81 & 0.81 & 0.00 & 0.50 & 0.00 & 0.00 & 0.02 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.01 & 0.00 & 0.00 & 0.01 & 0.10 & 0.31 & 0.37 & 0.26 \\ 0.01 & 0.01 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.01 & 0.23 & 0.67 & 0.81 & 0.57 \\ 0.01 & 0.01 & 0.00 & 0.00 & 0.02 & 0.00 & 0.00 & 0.01 & 0.31 & 0.93 & 1.12 & 0.79 \\ 0.00 & 0.00 & 0.08 & 0.13 & 0.04 & 0.13 & 0.13 & 0.00 & 0.32 & 0.72 & 0.86 & 0.61 \end{bmatrix}$$

$W =$

	Topic 1	Topic 2	Topic 3
d1	0.65	0.17	0.00
d2	0.48	2.20	0.04
d3	1.52	0.37	0.00
d4	2.37	0.00	0.00
d5	0.00	1.59	0.00
d6	0.01	0.00	0.55
d7	0.02	0.00	1.22
d8	0.02	0.00	1.69
d9	0.00	0.26	1.30

$H =$

	human	interface	computer	user	system	response	time	EPS	survey	trees	graph	minors
Topic 1	0.36	0.25	0.07	0.13	0.75	0.00	0.00	0.46	0.00	0.01	0.00	0.00
Topic 2	0.00	0.01	0.29	0.52	0.16	0.51	0.51	0.00	0.32	0.00	0.00	0.01
Topic 3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.19	0.55	0.66	0.47

Per fare un confronto con la decomposizione SVD, viene nuovamente preso in considerazione il documento numero 4 che, come si nota dalla matrice W, è fortemente correlato al primo topic. Quest'ultimo è definito maggiormente delle parole *system*, *EPS* e *Human*, esattamente come nel caso modellato dall'Analisi Semantica Latente. Quindi, abbiamo visto che entrambi i metodi associano al documento 4 un argomento che può essere interpretato come un sistema di user interface; tuttavia l'analisi NMF in alcuni contesti è preferibile proprio perché offre una rappresentazione più chiara e intuiva.

Capitolo 5

Probabilistic Latent Semantic Analysis (PLSA)

Il Probabilistic Latent Semantic Analysis (PLSA), proposto nel 1999 da Hofmann [26], rappresenta un'evoluzione diretta di LSA, introducendo un modello probabilistico invece della fattorizzazione SVD per affrontare il problema del topic modeling.

5.1 Modello generativo probabilistico

Chiaramente l'attività di scrittura di un testo è un processo complesso in cui l'autore sceglie accuratamente le parole con lo scopo di esporre il meglio possibile uno o più argomenti.

Un modello generativo si fonda su questa idea, assumendo che l'uso dei termini che compongono un documento dipenda in modo diretto da una struttura latente che specifica i temi. Ad esempio, prendendo in considerazione un articolo scientifico, si suppone che lo scrittore per primo scelga un argomento latente, come "l'energia nucleare", e poi selezioni delle parole specifiche associate a quel argomento secondo una certa distribuzione di probabilità.

Tramite tale assunzione, i modelli generativi cercano di replicare il processo di produzione di un documento semplificandolo in un insieme di passaggi probabilistici che possono essere formalizzati a livello matematico. I testi generati seguiranno la rappresentazione Bag of Words 3.2.2 e quindi saranno incomprensibili in quanto il loro significato, derivante dalla sequenza logica delle parole, verrà perso. L'idea di base è che sia comunque possibile distinguere gli argomenti di un testo anche con una permutazione casuale delle parole dello stesso.

Grazie alla costruzione di un modello che descrive come i dati osservati potrebbero essere generati, possiamo invertire il processo sfruttando delle tecniche statistiche: dato un insieme di documenti, possiamo cercare di risalire alla struttura latente che più probabilmente ha prodotto quei dati. In questo modo arriviamo ad ottenere l'obiettivo del topic modeling, ossia

identificare gli argomenti che vengono affrontati nei documenti.

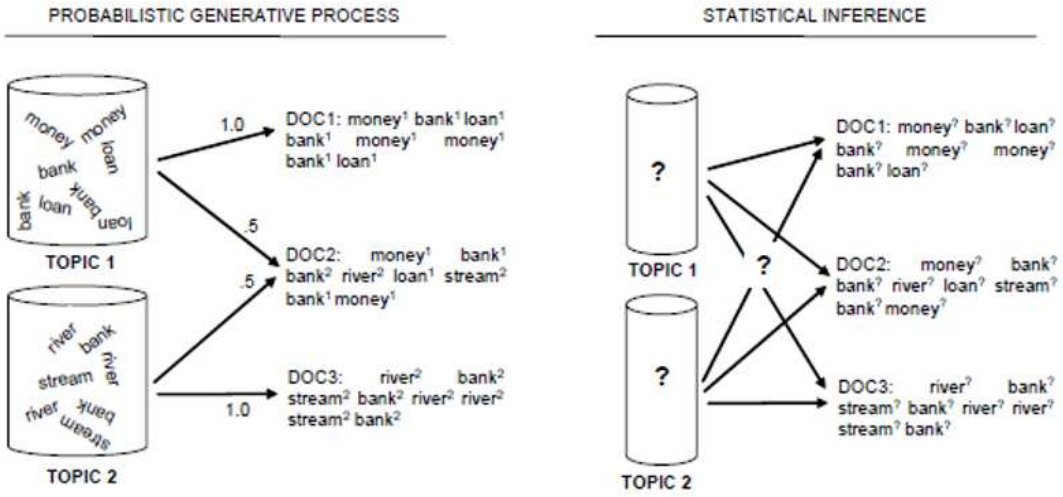

Figura 5.1: Processo generativo e problema di inferenza statistica [27]

5.1.1 Fasi del modello

In sintesi, possiamo descrivere il modello generativo probabilistico come la successione delle seguenti fasi:

1. Input: la fase iniziale prevede la suddivisione dei testi in token di parole con l'intenzione di costruire una matrice documento-termine di riferimento.

2. Processo generativo: attraverso un modello probabilistico viene simulata la creazione del corpus. Il testo generato segue la rappresentazione Bow che, di fatto, prevede la formazione di un'altra matrice documento-termine.

3. Stima dei parametri (inferenza): si tratta della fase fondamentale del processo in quanto l'obiettivo è trovare i parametri del modello che meglio permettono la realizzazione di una BoW simile alla matrice documento-termine iniziale. Dunque, questo processo di stima si basa proprio sul confronto tra la matrice documento-termine in input e quella generata dal modello e, in generale, è eseguito tramite algoritmi di ottimizzazione iterativa, che continuano ad aggiornare i parametri del modello fino alla convergenza di valori che spiegano al meglio i dati di partenza osservati.

4. Output: vengono restituiti i parametri del modello trovati nel passaggio precedente:

- θ_d che rappresenta la distribuzione degli argomenti trattati nel documento d, con $d \in \{1, \ldots, j, \ldots, M\}$ e M è il numero totale di documenti. Nello specifico, ogni documento è associato ad una distribuzione di probabilità su T temi tale che $\theta_d = [\theta_{d,1}, \theta_{d,2}, \ldots, \theta_{d,T}]$ e $\sum_{k=1}^{T} \theta_{d,k} = 1$. Per indicare l'insieme delle distribuzioni definite per tutti i documenti si usa $\theta_{1:M}$.
- ϕ_z che rappresenta la distribuzione delle parole del vocabolario per l'argomento z, con $z \in \{1, \ldots, k, \ldots, T\}$ e T è il numero totale di temi. Ogni argomento è associato ad una distribuzione di probabilità su V parole, tale che $\phi_z = [\phi_{z,1}, \phi_{z,2}, \ldots, \phi_{z,V}]$ e $\sum_{w=1}^{V} \phi_{z,w} = 1$. Con $\phi_{1:T}$ si indica l'insieme di tutte le distribuzioni definite per gli argomenti.

5.2 La distribuzione multinomiale

Come illustrato, i documenti vengono definiti come una distribuzione sui topic, mentre gli argomenti come una distribuzione sulle parole. Questi parametri vengono modellati nel processo generativo tramite una distribuzione multinomiale.

La distribuzione multinomiale è una distribuzione discreta che, per essere compresa al meglio, può essere messa in relazione all'esempio pratico di un dado truccato: il numero di facce è la dimensione della distribuzione, mentre le facce stesse, caratterizzate ciascuna da una probabilità differente di essere estratta, rappresentano i parametri.

La formula della funzione di probabilità è la seguente [28]:

$$f(x_1, x_2, \ldots, x_k) = \frac{n!}{x_1!\, x_2! \cdots x_k!}\, \theta_1^{x_1}\, \theta_2^{x_2} \cdots \theta_k^{x_k} \tag{5.1}$$

Dove:

- n è il numero totale di prove.
- k è il numero di esiti possibili, detti categorie.
- x_i rappresenta il numero di volte che l'esito di indice i si verifica.
- θ_i è la probabilità relativa all'esito di indice i, tale che $\sum_i \theta_i = 1$.

Ritornando all'esempio del dado, immaginando che abbia sei facce ($K = 6$) con probabilità $\theta_1, = 10/30$ e $\theta_2, \ldots, \theta_6 = 4/30$, possiamo calcolare la probabilità che su 5 lanci ($n = 5$) vengano estratte due volte la faccia 1, una volta la faccia 3 e due volte la faccia 6. In tal caso si applicherebbe la formula:

$$f(x_1 = 2, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 0, x_6 = 2) = \frac{10!}{2!\,0!\,1!\,0!\,0!\,2!} \left(\frac{10}{30}\right)^2 \left(\frac{4}{30}\right)^3$$

La distribuzione multinomiale, di fatto, non è altro che la generalizzazione a più esiti, ossia a più categorie, della distribuzione binomiale. In modo simile, a sua volta, la distribuzione binomiale estende la distribuzione bernoulliana a più prove. Segue una breve spiegazione.

La distribuzione di Bernoulli, ovvero la più semplice delle tre, descrive un esperimento caratterizzato da due esiti possibili: successo (con probabilità θ) o fallimento (con probabilità $1 - \theta$). La relativa funzione di probabilità $f(x) = \theta^x(1 - \theta)^{1-x}$, dove x può assumere solo i valori zero o uno, è utile per descrivere la probabilità di eventi binari come per esempio il lancio di una moneta. La distribuzione binomiale ha introdotto il numero prove n, estendendo così un esperimento di Bernoulli a più eventi indipendenti. Quindi la funzione di probabilità binomiale, $f(x) = \binom{n}{x}\theta^x(1 - \theta)^{n-x}$, permette di descrivere la probabilità di ottenere un certo numero x di successi su n lanci (prove). Infine, la distribuzione multinomiale invece di permettere solo due esiti possibili, ovvero successo e fallimento, estende la distribuzione multinomiale a k categorie differenti, come nel caso di un dado a k facce.

Per quanto concerne il processo generativo di PLSA e LDA (sezione 6), i testi verranno prodotti attraverso l'estrazione successiva di topic e parole a partire da due distribuzioni multinomiali di parametri θ_d e ϕ_z. Ad esempio, l'estrazione di un topic $z \sim \text{Multinomial}(\theta_d)$ avviene seguendo la formula :

$$\prod_i \theta_i^{1[k=i]} \tag{5.2}$$

dalla quale, ipotizzando di avere il parametro semplificato $\theta_d = \begin{bmatrix} 0.1 \\ 0.6 \\ 0.3 \end{bmatrix}$, la probabilità di estrarre il topic $k = 2$ è uguale a 0.6.

5.3 Processo generativo

L'idea di base, come detto precedentemente, è quella di sviluppare un processo probabilistico che si assume abbia generato i dati che stiamo osservando, cioè il corpus di documenti. Viene costruita una narrazione stocastica che permette di specificare come le variabili osservate vengano generate per mezzo delle variabili latenti: ogni variabile viene associata ad una distribuzione evidenziando le dipendenze che sussistono con i parametri e le altre variabili del

modello.

Il modello PLSA che genera i testi dei documenti può essere cosi descritto:

1. Selezione di un documento d_j con probabilità $P(d_j)$.
2. Per ogni parola nel documento:
   (a) Viene selezionato un argomento z_k con probabilità $P(z_k|d_j)$, $z \sim$ Multinomial(θ_{d_j}).
   (b) Viene selezionata una parola w_i con probabilità $P(w_i \mid z_k)$, $w \sim$ Multinomial(ϕ_{z_k}).

5.3.1 Modello grafico probabilistico

Un modello grafico probabilistico è una rappresentazione visiva che, attraverso un grafo aciclico diretto, permette di esprimere in modo chiaro ed intuitivo le dipendenze condizionali tra le variabili aleatorie del sistema. In particolare, i nodi del grafo rappresentano le variabili del modello; gli archi invece rappresentano le relazioni di dipendenza statistica tra le variabili. La distribuzione di un nodo dipende in modo esclusivo dai genitori, mentre i nodi non connessi sono da intendersi condizionatamente indipendenti. Le variabili con più istanze vengono rappresentate in un rettangolo, detto plate, che riporta nella parte inferiore il numero di replicazioni: esse si dicono scambiabili poiché se vengono applicate permutazioni la relativa distribuzione congiunta rimane invariata.

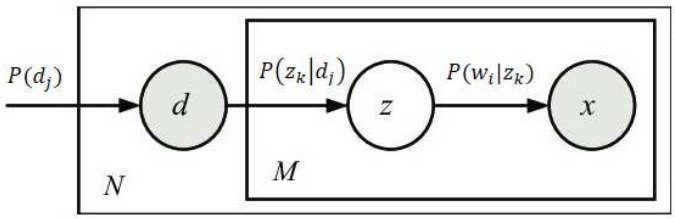

Figura 5.2: Modello grafico probabilistico PLSA

5.3.2 Distribuzione congiunta

La distribuzione congiunta è una funzione che permette di descrivere la probabilità delle variabili del modello, ed in generale costituisce una rappresentazione probabilistica completa necessaria per poter definire le procedure di inferenza che stimano le variabili latenti tramite le osservazioni.

L'espressione della probabilità congiunta si fonda su due assunzioni: la scambiabilità dei documenti e la scambiabilità delle parole. La prima ipotesi implica che l'ordine dei documenti all'interno del corpus non sia rilevante, in modo tale che sia possibile trattare ogni documento indipendentemente dagli altri, definendo la probabilità congiunta come il prodotto delle probabilità individuali di ciascun documento. La seconda invece non è altro che l'assunzione Bag-of-Words definita precedentemente (3.2.2): ogni documento viene considerato come una borsa di parole in cui l'ordine sequenziale di come queste appaiono viene ignorato.

Formalmente, la probabilità congiunta di vedere una generica parola w_i in un dato documento d_j è data da:

$$P(d_j, w_i) = P(d_j)P(w_i \mid d_j) \tag{5.3}$$

$$P(w_i \mid d_j) = \sum_{k=1}^{T} P(w_i \mid z_k)P(z_k \mid d_j) \tag{5.4}$$

$$P(d_j, w_i) = P(d_j) \sum_{k=1}^{T} P(w_i \mid z_k)P(z_k \mid d_j) \tag{5.5}$$

5.4 Relazione con LSA

Facendo un parallelismo con LSA, si vuole sviluppare un modello tale che $P(d_j, w_i)$, per ogni documento d_j e parola w_i, corrisponde alla stessa voce della matrice A documento-termine [19]. Tale probabilità è espressa dall'equazione esposta nella sezione precedente:

$$P(d_j, w_i) = P(d_j) \sum_{k=1}^{T} P(w_i \mid z_k)P(z_k \mid d_j) \tag{5.6}$$

$$P(d_j, w_i) = \sum_{k=1}^{T} P(w_i \mid z_k)P(d_j \mid z_k)P(z_k) \tag{5.7}$$

Utilizzando la regola di Bayes otteniamo [29]:

$$P(z_k \mid d_j) = \frac{P(d_j \mid z_k)P(z_k)}{P(d_j)} \tag{5.8}$$

$$P(z_k \mid d_j)P(d_j) = P(d_j \mid z_k)P(z_k) \tag{5.9}$$

$$P(w_i \mid z_k)P(z_k \mid d_j)P(d_j) = P(w_i \mid z_k)P(d_j \mid z_k)P(z_k) \tag{5.10}$$

$$P(d_j)\sum_{k=1}^{T} P(w_i \mid z_k)P(z_k \mid d_j) = \sum_{k=1}^{T} P(w_i \mid z_k)P(d_j \mid z_k)P(z_k) \tag{5.11}$$

$$P(d_j)\sum_{k=1}^{T} P(w_i \mid z_k)P(z_k \mid d_j) = \sum_{k=1}^{T} P(z_k)P(d_j \mid z_k)P(w_i \mid z_k) \tag{5.12}$$

$$P(d_j, w_i) = \sum_{k=1}^{T} P(z_k)P(d_j \mid z_k)P(w_i \mid z_k) \tag{5.13}$$

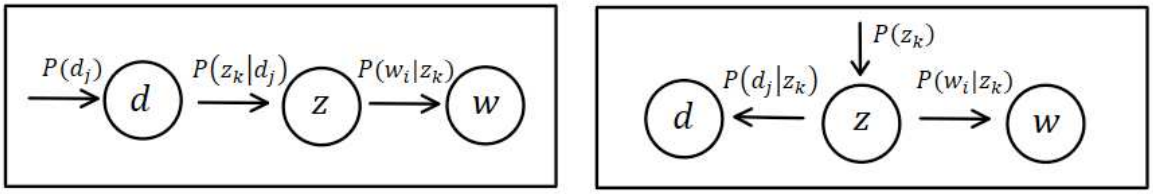

Figura 5.3: Rappresentazione asimmetrica e simmetrica di PLSA

Questa nuova parametrizzazione permette di mettere a confronto diretto il modello PLSA con LSA:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$$

La probabilità di un argomento $P(z_k)$ corrisponde alla matrice diagonale che contiene le probabilità degli argomenti, la probabilità di un documento dato l'argomento $P(d_j \mid z_k)$ corrisponde alla matrice documento-argomento U, ed infine, la probabilità della nostra parola dato l'argomento $P(w_i \mid z_k)$ corrisponde alla matrice termine-argomento V.

5.5 Stima dei parametri

L'obiettivo adesso è determinare i parametri del modello a partire dai dati a disposizione. Uno degli approcci più classici prevede la definizione della funzione di verosimiglianza che indica quanto i dati osservati (matrice documento-termine in input), siano verosimili per dei determinati parametri. Nel modello PLSA, i parametri sono le due distribuzioni condizionali $P(w_i \mid z_k)$ e $P(z_k \mid d_j)$, regolate dai vettori di probabilità θ_{d_j} e ϕ_{z_k}. Dunque, stimare i parametri del modello implica trovare tutti i vettori $\phi_{1:T}$ e $\theta_{1:M}$ che massimizzano la verosimiglianza dei dati osservati.

5.5.1 La funzione di verosimiglianza

La funzione di probabilità e la funzione di verosimiglianza sono formalmente identiche ma si caratterizzano per perseguire scopi differenti. La funzione di probabilità infatti ha i parametri θ noti ed ha come incognita la probabilità delle variabili aleatorie x_i; la funzione di verosimiglianza invece ha noto le variabili x_i, ed è deputata a calcolare quali parametri θ siano più verosimili in relazione alle variabili osservate [30].

La funzione di verosimiglianza si indica comunemente tramite utilizzo della lettera L, come segue:

$$L(x_1, \ldots, x_n \mid \theta) = f(x_1, \ldots, x_n \mid \theta)$$

Un'altra differenza che contraddistingue la funzione di verosimiglianza è che non ha area sottesa unitaria: per questo motivo, non permette di fornire misure di probabilità assolute, ma solamente informazioni che indicano la plausibilità dei diversi valori che possono assumere i parametri alla luce dei dati osservati.

A fini pratici, è comunemente utilizzato il logaritmo naturale della verosimiglianza:

$$l(\theta) = \log L(\theta) \tag{5.14}$$

Questo è possibile in quanto $L(\theta)$ e $l(\theta)$ hanno punti di massimo in corrispondenza degli stessi valori di θ. La motivazione principale legata a questa scelta, è che la verosimiglianza corrispondente al prodotto di numeri a bassa probabilità può assumere un valore molto piccoli, anche dell'ordine di 10^{-34}. In questi casi, a causa degli arrotondamenti numerici adottati da parte dei calcolatori, verrebbero prodotti molti errori compromettendo i risultati. Tali problematicità, grazie alle proprietà del logaritmo, vengono evitate poiché i prodotti vengono sostituti da somme rendendo i valori più gestibili in termini di dimensione. Inoltre, si ricordi che lo scopo è trovare il valore $\hat{\theta}$ che massimizza la funzione di verosimiglianza, che è derivabile ovunque rispetto a θ e non ha massimo coincidente con un estremo del campo di variazione di θ. Dunque la trasformazione dei prodotti in somma, permette anche di facilitare i calcoli legati alla determinazione del valore per cui la derivata prima si annulla rispetto a θ [31].

$$\frac{dl(\theta \mid \mathbf{x})}{d\theta} = 0$$

5.5.2 Funzione di verosimiglianza per PLSA

Nel caso di PLSA la funzione di verosimiglianza descrive quanto è probabile osservare un determinato insieme di parole, cioè un documento, dati i parametri del modello, ovvero le distribuzioni $P(w_i \mid z_k)$ e $P(z_k \mid d_j)$ che sono regolate dalle matrici $\theta_{1:M}$ e $\phi_{1:T}$.

La funzione di verosimiglianza da massimizzare è data da [32]:

$$L = \sum_{j=1}^{M} \sum_{i=1}^{V} n(d_j, w_i) \log P(d_j, w_i) \tag{5.15}$$

Dove:

- $n(d_j, w_i)$ rappresenta il numero di volte in cui il termine w_i è apparso nel documento d_i.
- $P(d_j, w_i)$ rappresenta la probabilità congiunta di osservare la parola w_j nel documento d_i definita dall'equazione 5.4.

Per permettere calcoli più semplici essa viene definita attraverso il logaritmo naturale, ottenendo:

$$L = \sum_{j=1}^{M} n(d_j) \left[\log P(d_j) + \sum_{i=1}^{V} \frac{n(d_j, w_i)}{n(d_j)} \log \sum_{k=1}^{T} P(w_i \mid z_k) P(z_k \mid d_j) \right] \tag{5.16}$$

Qui $n(d_j) = \sum_i n(d_j, w_i)$ si riferisce alla lunghezza del documento.

5.5.3 Ottimizzazione con l'algoritmo Expectation-Maximization (EM)

Per quanto riguarda la massimizzazione della verosimiglianza, a causa delle somme all'interno del logaritmo, il calcolo delle derivate parziali della funzione non è comunque facilmente praticabile. Per questo motivo si utilizzano delle tecniche di stima come l'algoritmo *Expectation Maximization (EM)*, che alterna due passaggi: l'Expectation step (E) in cui, sulla base delle stime correnti dei parametri, vengono calcolate le probabilità a posteriori delle variabili latenti, e il Maximization step (M) in cui i parametri vengono aggiornati per massimizzare la verosimiglianza attesa che dipende dalle probabilità a posteriori ottenute nel passaggio precedente [33].

1. Fase E: attraverso l'applicazione della formula di Bayes vengono calcolate le distribuzioni a posteriori $P(z_k \mid d_j, w_i)$ a partire dalla stima precedente dei parametri:

$$P(z_k \mid d_j, w_i) = \frac{P(z_k \mid d_j) P(w_i \mid z_k)}{\sum_{l=1}^{T} P(z_t \mid d_j) P(w_i \mid z_t)} \tag{5.17}$$

2. Fase M: i parametri $P(w_i \mid z_k)$ e $P(z_k \mid d_j)$ vengono aggiornati per massimizzare la verosimiglianza utilizzando le probabilità a posteriori $P(z_k \mid d_j, w_i)$. Nello specifico, le equazioni finali che si svolgono sono.

$$P(w_i \mid z_k) = \frac{\sum_{i=1}^{V} n(d_j, w_i) P(z_k \mid d_j, w_i)}{\sum_{v=1}^{V} \sum_{j=1}^{M} n(d_j, w_v) P(z_k \mid d_j, w_v)} \tag{5.18}$$

$$P(z_k \mid d_j) = \frac{\sum_{j=1}^{M} n(d_j, w_i) P(z_k \mid d_j, w_i)}{\sum_{i=1}^{V} n(d_j, w_i)} \tag{5.19}$$

I passaggi E e M vengono iterati fino alla realizzazione di una condizione di terminazione. In generale, è possibile utilizzare la tecnica di early stopping in cui non si ottimizza fino alla convergenza totale dei parametri al fine di evitare l'overfitting, ovvero una situazione in cui il modello si adatta in modo eccessivo ai dati di addestramento perdendo capacità di generalizzazione [32].

Capitolo 6

Latent Dirichlet allocation (LDA)

Il Latent Dirichlet Allocation (LDA) è un modello introdotto nel 2003 da Blei et al. [34] che, come PLSA, semplifica il processo di creazione di un documento in una serie di passaggi probabilistici e definisce una distribuzione di probabilità dei topic su documenti del corpus.

6.1 Approccio Bayesiano

La statistica classica prevede la stima dei parametri di un modello solo attraverso l'osservazione dei dati; l'approccio bayesiano ha permesso un cambio di paradigma definendo un processo che utilizza anche le informazioni a priori, ovvero le conoscenze o ipotesi precedenti alle osservazioni. In questa tecnica statistica, attraverso l'applicazione della regola di Bayes 6.1, le conoscenze a priori vengono modificate progressivamente in relazione ai nuovi dati osservati. In questo modo, è possibile ottenere valori aggiornati delle probabilità a posteriori, che rappresentano le convinzioni sull'incertezza dei parametri dopo aver considerato nuove prove.

La regola di Bayes è definita come segue:

$$P(\theta \mid X) = \frac{P(X \mid \theta)\, P(\theta)}{P(X)} \tag{6.1}$$

Dove:

- $P(\theta \mid X)$: è detta Posterior e rappresenta la probabilità a posteriori di θ. Nello specifico, esprime le convinzioni sul parametro θ una volta che sono state prese in considerazioni le osservazioni delle prove X.

- $P(\theta)$: è chiamata probabilità a priori (Prior) e definisce le convinzioni iniziali sul parametro θ senza tener conto delle prove X.

- $P(X \mid \theta)$: prende il nome di verosimiglianza e rappresenta la probabilità di vedere le variabili X dato un certo valore assunto dal parametro θ.

- $P(X)$: è detta evidenza o probabilità marginale di X. In particolare, questa componente è definita come la probabilità complessiva dei dati osservati X, calcolata valutando il seguente integrale su tutti i valori di θ possibili.

$$P(X) = \int P(X \mid \theta) \cdot P(\theta)\, d\theta$$

  Nel concreto, è un termine che funge da costante di normalizzazione per la probabilità a posteriori $P(\theta \mid X)$, assicurandosi che sommi correttamente ad uno.

La regola di Bayes non è sempre applicabile a causa della difficoltà nel calcolare analiticamente l'integrale per il fattore di normalizzazione. Spesso, questo accade poiché i modelli richiedono un elevato numero di parametri che implicano la formazione di un integrale che deve essere valutato in uno spazio dimensionale potenzialmente molto grande. Per far fronte a questa problematica si utilizzano delle tecniche di inferenza approssimata che permettono di rendere i modelli bayesiani di dimensioni elevate trattabili.

In generale, l'intero processo bayesiano prevede [35]:

1. Formulazione delle credenze a priori: inizialmente, prima dell'osservazione dei dati, vengono rappresentate le nostre convinzioni a priori su un parametro di interesse per mezzo di una distribuzione di probabilità.

2. Analisi dati sperimentali: in questa fase vengono raccolti i dati del fenomeno in esame. Quindi viene formulata una funzione di verosimiglianza che indica quanto i dati appena osservati siano plausibili rispetto alle credenze sul parametro presupposte.

3. Aggiornamento delle credenze a posteriori: viene applicato il teorema di Bayes per combinare le informazioni acquisite dai dati alle convinzioni sviluppate a priori. In questo modo si ottiene una distribuzione a posteriori, ovvero le credenze sul parametro dopo aver preso in considerazione le osservazioni dei dati.

4. Inferenza: viene utilizzata la distribuzione a posteriori per stimare il parametro di interesse.

6.1.1 Distribuzioni coniugate

Dato un fenomeno, dobbiamo trovare una distribuzione di probabilità che ci permetta di quantificare le nostre credenze a priori sui parametri. Di solito, si sceglie una distribuzione a priori coniugata, ovvero di una speciale categoria che, se associata ad un particolare tipo di funzione di verosimiglianza, fornisce una distribuzione a posteriori della stessa famiglia della distribuzione a priori. In questo modo, si ottengono Prior e Posterior appartenenti alla stessa famiglia di distribuzione di probabilità, ma caratterizzati da parametri diversi. La scelta delle distribuzioni a priori coniugate non costituiscono un obbligo poiché le tecniche di inferenza consentono una vasta gamma di distribuzioni, ma rappresentano la migliore scelta in termini di calcoli matematici. Un altro vantaggio, ottenendo Prior e Posterior dello stesso tipo, è che sarà possibile usare la regola di Bayes iterativamente per perfezionare le proprie convinzioni ogni volta che si osservano nuovi dati [30].

6.1.2 Esempio

Chiaramente esistono diverse distribuzioni a priori coniugate; si citano ad esempio i modelli Poisson-gamma, Normale-Normale, uniforme-Pareto, Beta-binomiale e il modello Dirichlet-multinomiale, che viene utilizzato proprio in LDA.

Di seguito viene riportato un esempio pratico del processo bayesiano che utilizza lo schema Beta-binomiale. Il fine è mostrare un caso esemplificativo di questo paradigma statistico e presentare la distribuzione Beta, che rappresenta la base per comprendere la distribuzione Dirichlet.

Distribuzione Beta La distribuzione Beta viene utilizzata per descrivere fenomeni il cui insieme di possibili valori è limitato all'intervallo aperto $(0,1)$. In particolare, si dice che una variabile aleatoria $\theta \in (0,1)$ segue la distribuzione Beta di parametri (α, β) se la sua densità è:

$$Beta(\theta \mid \alpha, \beta) = \begin{cases} \frac{1}{B(\alpha,\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}, & \text{se } 0 < \theta < 1 \\ 0, & \text{altrimenti.} \end{cases}$$

Dove:

- $\theta \in (0,1)$ è la variabile aleatoria della distribuzione che rappresenta una probabilità.
- I parametri $\alpha > 0$ e $\beta > 0$ influenzano la forma della distribuzione (figura 6.1) e possono essere interpretati come le credenze a priori associate ad una sequenza di prove binarie: α rappresenta il numero di "successi", mentre β il numero di "fallimenti".

- $B(\alpha, \beta)$ è la funzione Beta che nella distribuzione funge da costante di normalizzazione con lo scopo di rendere l'integrale della densità uguale ad 1. La funzione Beta è data da:

$$B(\alpha, \beta) = \int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1}\, d\theta = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} \tag{6.2}$$

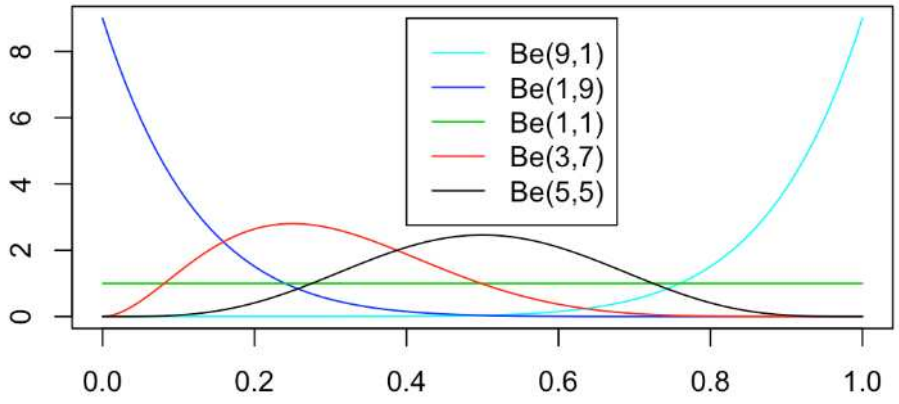

Figura 6.1: Distribuzione Beta [30]

La distribuzione Beta, proprio perché limitata nell'intervallo aperto $(0,1)$, escludendo i casi limite, è particolarmente adatta a modellare fenomeni descritti da percentuali o proporzioni, come la probabilità di successo di un evento binomiale.

Lancio di una moneta Il primo passo per analizzare un lancio di una moneta tramite approccio bayesiano, come menzionato precedentemente, è definire le nostre convinzioni iniziali riguardo all'equità della moneta, ovvero la probabilità θ di avere un successo (estrarre testa): viene quindi definita una distribuzione $\theta \sim \text{Beta}(\alpha, \beta)$.

Ad esempio, scegliendo una distribuzione a priori $\theta \sim \text{Beta}(\alpha = 4, \beta = 4)$, si suppone che ottenere croce sia probabile tanto quanto avere testa. Il valore $\theta = 0.5$, infatti, è il più probabile nella distribuzione, ma vengono comunque ritenuti possibili anche tutti gli altri valori del parametro, estremi esclusi (non sono modellate le casistiche in cui si ottiene solo testa o solo croce). Chiaramente questi parametri, in base alla sicurezza delle convinzioni, si possono modificare per accentuare o meno tale caratteristica.

Il passo successivo prevede la raccolta dei dati sperimentali: ossia si effettuano N lanci della moneta di cui w hanno prodotto testa. Quindi, sfruttando la distribuzione binomiale, si sviluppa una funzione di verosimiglianza che esprime la probabilità di osservare i successi w in funzione del valore di θ.

$$p(w \mid \theta) = \binom{N}{w} \theta^w (1-\theta)^{N-w}$$

Quindi, usando la regola di Bayes, viene aggiornata la distribuzione a priori Beta con i dati osservati ottenendo la distribuzione a posteriori.

$$P(\theta \mid w) = \frac{P(w \mid \theta)\, P(\theta)}{P(w)}$$

In questo caso, avendo utilizzato la distribuzione di verosimiglianza $\text{Bin}(N, w \mid \theta)$ e la distribuzione a priori $\text{Beta}(\alpha, \beta)$, è possibile semplificare i calcoli utilizzando un teorema che stabilisce la distribuzione a posteriori del parametro θ uguale ad una distribuzione $(\theta \mid z) \sim \text{Beta}(\alpha + w, \beta + N - w)$.

Determinata la distribuzione a posteriori, si può fare inferenza sull'equità della moneta. Ad esempio, una stima del parametro θ è possibile ricavarla dal valore atteso della distribuzione a posteriori che in questo caso è definito come [30]:

$$E[\theta \mid w] = \frac{\alpha + w}{\alpha + \beta + N}$$

6.1.3 Nel contesto del topic modeling

Ritornando al contesto del topic modeling, esattamente come facevamo per PLSA, il nostro scopo è trovare i parametri del modello (θ_d e ϕ_z) a partire dai dati iniziali a disposizione (matrice documento-termine in input). Il modello PLSA, tuttavia, non segue completamente un approccio bayesiano; questi infatti non impone una distribuzione a priori sui parametri ma utilizza solamente tecniche di stima che massimizzano la verosimiglianza, come l'algoritmo EM spiegato nella sezione 5.5.3. Il modello LDA, invece, sfrutta il processo bayesiano utilizzando la distribuzione di Dirichlet per quantificare le proprie credenze a priori sui parametri. Viene infatti definita una distribuzione di Dirichlet come Prior sui parametri θ_d, ovvero la distribuzione dei topic trattati nei documenti, e sui parametri ϕ_z che rappresentano la distribuzione delle parole del vocabolario per ogni argomento. Poi, a livello teorico, LDA osserva i dati iniziali (matrice documento-termine in input) e calcola la funzione di verosimiglianza che descrive la probabilità di osservare quei dati per i parametri θ_d e ϕ_z. Tuttavia, come viene analizzato nella sezione 6.4, il calcolo esatto della distribuzione a posteriori non è sempre trattabile computazionalmente a causa della complessità del modello. Per far fronte a questa problematica, LDA deve ricorrere a tecniche di inferenza approssimata, come ad esempio il Gibbs Sampling, che permettono tramite progressive ottimizzazioni di approssimare la distribuzione a posteriori, consentendo di stimare i parametri θ_d e ϕ_z e trovare gli argomenti

trattati nei documenti.

6.2 La distribuzione di Dirichlet

La distribuzione di Dirichlet generalizza la distribuzione Beta per k categorie in modo simile a come fatto dalla distribuzione multinomiale per la binomiale (sezione 5.2). Come mostrato, la distribuzione Beta permette di modellare proporzioni per eventi binari, ad esempio la probabilità di successo θ nel lancio di una moneta. La distribuzione di Dirichlet estende la restrizione dei due esiti possibili, ovvero successo e fallimento, permettendo di trattare fenomeni che prevedono k casistiche differenti, come un dado a k facce. In sintesi, Beta e Dirichlet sono dette distribuzioni di distribuzioni: la prima modella una probabilità binomiale, mentre la seconda un vettore di probabilità multinomiale di dimensione k.

La funzione di distribuzione di probabilità è definita dalla seguente equazione [28]:

$$\text{Dir}(\vec{\theta} \mid \vec{\alpha}) = \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1} \tag{3.5}$$

In cui il primo termine può essere espresso tramite la funzione Beta, come mostrato:

$$\frac{1}{B(\alpha)} = \frac{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}{\prod_{i=1}^{K} \Gamma(\alpha_i)}$$

Ottenendo una somiglianza diretta con la distribuzione Beta.

$$\text{Beta}(\theta \mid \alpha, \beta) = \frac{1}{B(\alpha, \beta)} \theta^{\alpha - 1} (1 - \theta)^{\beta - 1}$$

$$\text{Dir}(\vec{\theta} \mid \vec{\alpha}) = \frac{1}{B(\alpha)} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1} \tag{6.3}$$

Qui:

- $\vec{\theta} = (\theta_1, \ldots, \theta_k)$ è il vettore delle variabili casuali della distribuzione tale che $\theta_i \geq 0 \; \forall i$ e $\sum_{i=1}^{K} \theta_i = 1$. Rappresenta di fatto un vettore di probabilità in cui ogni θ_i è associato alla probabilità della categoria i.

- $\vec{\alpha} = (\alpha_1, \ldots, \alpha_k)$ è il vettore che parametrizza la distribuzione. Questi valori permettono di modellare la forma della distribuzione regolando l'intensità con cui ciascuna categoria prevale.

- $B(\alpha, \beta)$ è la funzione Beta che, come nel caso precedente, agisce da costante di normalizzazione assicurandosi che la distribuzione integri ad 1 sul dominio.

La distribuzione di Dirichlet può essere interpretata geometricamente come una distribuzione di probabilità definita su un simplesso $(k-1)$-dimensionale. Per $k = 3$ il supporto della distribuzione è un simplesso bidimensionale, ovvero un triangolo equilatero i cui vertici rappresentano le k categorie. I parametri $\vec{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$ permettono di modellare la forma della distribuzione: se sono minori di uno $(\alpha_i < 1)$ si ottengono dei picchi agli angoli del simplesso, se sono unitari $(\alpha_i = 1)$ si ottiene una distribuzione uniforme, ed infine se sono maggiori di uno $(\alpha_i > 1)$ la distribuzione tende verso centro del simplesso con intensità proporzionale ai valori dei parametri.

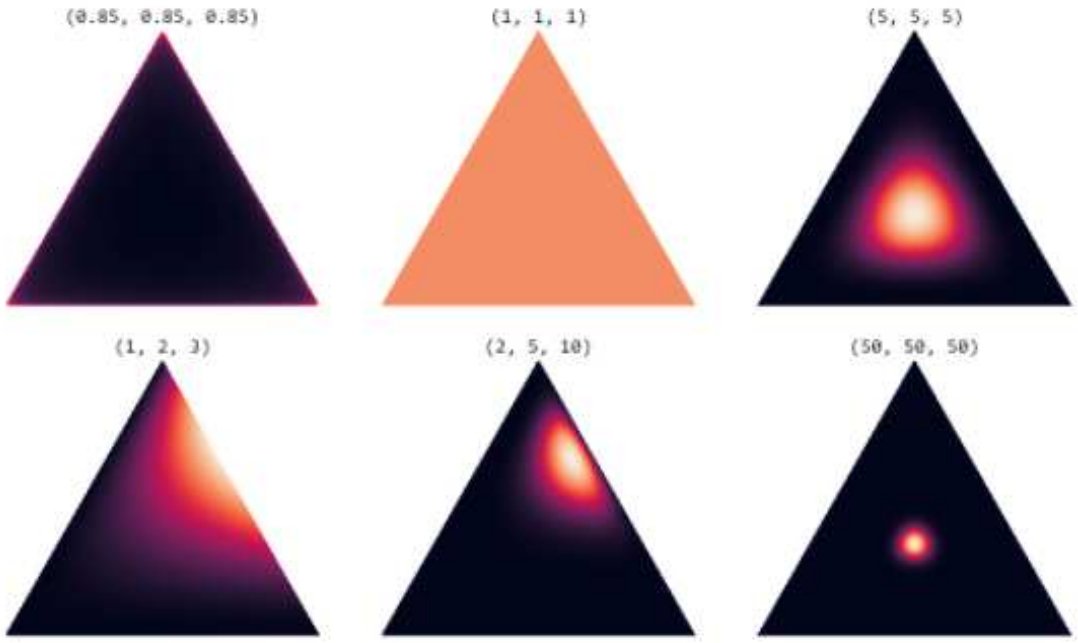

Figura 6.2: Distribuzione di Dirichlet su simplesso bidimensionale al variare dei parametri α_i [36]

Continuando il parallelismo tra la distribuzione Beta e Dirichlet, si consideri di avere come ipotesi una moneta equa, $\theta = 0.5$, e un dado equilibrato, $\vec{\theta} = (\theta_1, \ldots, \theta_6) = \left(\frac{1}{6} \ldots, \frac{1}{6}\right)$. Si è visto come nella distribuzione Beta sia possibile definire i parametri α e β per massimizzare la probabilità attorno a $\theta = 0.5$. Allo stesso modo, nella distribuzione di Dirichlet i parametri $\alpha_1, \ldots, \alpha_6$ possono essere stabiliti al fine di regolare la concentrazione della distribuzione ed accentuarla attorno al vettore di probabilità $\vec{\theta}$, in cui ciascuna categoria avrà circa la stessa

probabilità di essere estratta.

Nel contesto di LDA, vengono definite due tipologie differenti di distribuzioni di Dirichlet per esprimere le ipotesi a priori: una con parametro $\vec{\alpha}$ sulla distribuzione dei topic trattati nei documenti θ_d, e l'altra con parametro $\vec{\beta}$ sulla distribuzione delle parole per ogni argomento ϕ_z. Solitamente, per modellare l'assenza di una conoscenza a priori sulla probabilità dei topic, vengono utilizzate distribuzioni di Dirichlet simmetriche, ossia distribuzioni in cui i vettori $\vec{\alpha}$ e $\vec{\beta}$ hanno tutti gli elementi uguali ($\alpha = \{\alpha_1, \ldots, \alpha_T\} = \alpha_i$ e $\beta = \{\beta_1, \ldots, \beta_W\} = \beta_i$).

Generalmente, i parametri α assumono valori vicini allo zero, in modo che le probabilità tendano a concentrarsi su un numero ristretto di topic. I valori $\alpha < 1$, infatti, permettono di produrre vettori θ_d sparsi, ovvero in cui solo la probabilità di alcuni topic risalta. Questo è congruo con le aspettative poiché un documento in modo comune, tratta solamente di specifici settori senza spaziare su tutti gli argomenti possibili [37].

Per quanto riguarda i parametri β, invece, se assumono valori elevati allora si ottengono topic caratterizzati da più parole con alte probabilità, suggerendo la scelta di un numero basso di topic T. Per contro, se i parametri assumono valori ridotti si avranno poche parole con probabilità rilevanti, favorendo quindi un numero maggiore di argomenti. Anche in questo caso, di solito si tende a prediligere gli iperparametri che permettono di avere distribuzioni ϕ_z sparse, poiché in questo modo si creano topic più distinti essendo caratterizzati solo da poche parole rappresentative.

Distribution	Density	Example Parameters	Example Draws
Multinomiale	$\prod_i \phi_i^{\mathbf{1}[w=i]}$	$\phi = \begin{bmatrix} 0.1 \\ 0.6 \\ 0.3 \end{bmatrix}$	$w = 2$
Dirichlet	$\frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1}$	$\alpha = \begin{bmatrix} 1.1 \\ 0.1 \\ 0.1 \end{bmatrix}$	$\theta = \begin{bmatrix} 0.8 \\ 0.15 \\ 0.05 \end{bmatrix}$

Figura 6.3: Distribuzione multinomiale e distribuzione di Dirichlet [38]

6.3 Processo generativo

Il processo generativo del modello è descritto da questi passaggi:

1. Per ogni topic $t \in \{1, \ldots, k, \ldots, T\}$:

(a) A partire da una distribuzione di Dirichlet simmetrica si estrae il parametro di una distribuzione multinomiale del topic t sulle parole del vocabolario, $\phi_z \sim \text{Dir}(\vec{\beta})$

2. Per ogni documento $d \in \{1, \ldots, j, \ldots, D\}$:

   (a) A partire da una distribuzione di Dirichlet simmetrica si estrae il parametro di una distribuzione multinomiale del documento d sui topic, $\theta_d \sim \text{Dir}(\vec{\alpha})$

   (b) Per ogni parola $w \in \{1, \ldots, i, \ldots, N_d\}$

      i. Viene estratto un topic dalla distribuzione del documento d sui topic, $z_{dn} \sim \text{Multinomial}(\theta_d)$

      ii. Viene estratta una parola dalla distribuzione dei topic sulle parole, $w_{dn} \sim \text{Multinomial}(\phi_z)$

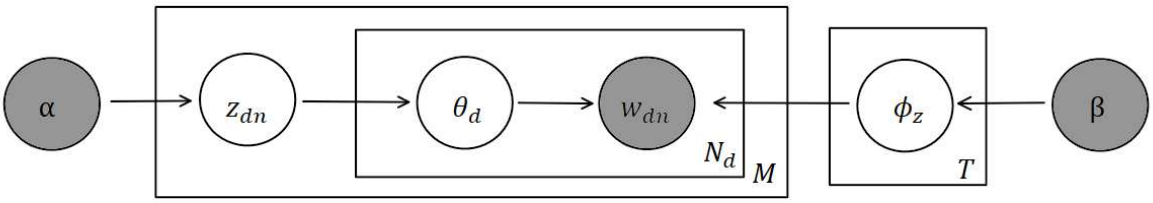

Figura 6.4: Modello grafico probabilistico LDA

6.3.1 Probabilità congiunta

La probabilità congiunta del modello LDA, assumendo sempre le ipotesi di scambiabilità dei documenti e scambiabilità delle parole, è espressa dalla seguente formula [39]:

$$p(\mathbf{z}, \mathbf{w}, \theta_{1:M}, \phi_{1:T} | \vec{\alpha}, \vec{\beta}) = \prod_{t=1}^{T} p(\phi_t \mid \vec{\beta}) \prod_{d=1}^{M} p(\theta_d \mid \vec{\alpha}) \prod_{n=1}^{N_d} p(z_{dn} \mid \theta_d) p(w_{dn} \mid z_{dn}, \phi_{1:T}) \quad (6.4)$$

$$= \left(\prod_{t=1}^{T} \frac{\Gamma(V\beta)}{\prod_{v=1}^{V} \Gamma(\beta_v)} \prod_{v=1}^{V} \phi_{t,v}^{\beta_v - 1} \right) \left(\prod_{d=1}^{M} \frac{\Gamma(T\alpha)}{\prod_{t=1}^{T} \Gamma(\alpha_t)} \prod_{t=1}^{T} \theta_{d,t}^{\alpha_t - 1} \right) \times \prod_{n=1}^{N_d} \left(\prod_{t=1}^{T} \theta_{d,t}^{1[z_{dn}=t]} \prod_{v=1}^{V} \phi_{z,v}^{1[w_{dn}=v]} \right) \quad (6.5)$$

Da cui possiamo distinguere le differenti componenti nel modo che segue:

- $p(\phi_z \mid \vec{\beta})$ e $p(\theta_d \mid \vec{\alpha})$ sono le distribuzioni di Dirichlet che esprimono le ipotesi a priori: la prima descrive la distribuzione delle parole per il topic t; la seconda invece descrive la distribuzione dei topic per il documento d.

- $p(z_{dn} \mid \theta_d)$ e $p(w_{dn} \mid z_{dn}, \phi_{1:T})$ sono distribuzioni multinomiali che specificano la verosimiglianza del modello, ovvero la probabilità dei dati osservati date le variabili latenti. Le due distribuzioni descrivono rispettivamente la probabilità di estrarre il topic z_{dn} dato il vettore θ_d, e la probabilità di osservare la parola w_{dn} associata al topic z_{dn} data la rispettiva distribuzione ϕ_z.

6.4 Inferenza

L'obiettivo centrale del modello LDA è rappresentato dall'inversione del processo generativo al fine di determinare le variabili latenti date le variabili osservate. Precisamente, le variabili latenti da determinare sono:

- z_{dn}: l'assegnazione degli argomenti a ciascuna parola nei documenti. Per ogni parola di un documento, vogliamo determinare a quale argomento è assegnata.

- $\theta_{1:M}$: le distribuzioni degli argomenti per ciascun documento.

- $\phi_{1:T}$: le distribuzioni delle parole per ciascun argomento.

Il problema inferenziale che deve essere risolto è quindi il calcolo della distribuzione a posteriori tramite la regola di Bayes:

$$p(\mathbf{z}, \theta_{1:D}, \phi_{1:T}|\mathbf{w}, \vec{\alpha}, \vec{\beta}) = \frac{p(\mathbf{z}, \mathbf{w}, \theta_{1:M}, \phi_{1:T}|\vec{\alpha}, \vec{\beta})}{p(\mathbf{w}|\vec{\alpha}, \vec{\beta})} \tag{6.6}$$

Questa distribuzione, tuttavia, non è trattabile analiticamente a causa della difficoltà di valutare l'evidenza, data dal seguente integrale ottenuto marginalizzando sulle variabili latenti.

$$p(\mathbf{w}|\vec{\alpha}, \vec{\beta}) = \sum_{z} \int_{S_{\theta_{1:D}}} \int_{S_{\phi_{1:T}}} p(\mathbf{z}, \mathbf{w}, \theta_{1:M}, \phi_{1:T}|\vec{\alpha}, \vec{\beta}) d\phi_{1:T} d\theta_{1:D}$$

Nonostante l'inferenza esatta sia intrattabile, è comunque possibile utilizzare una vasta gamma di algoritmi di inferenza approssimata che possono essere applicati al problema in modo efficace, tra cui: la *massimizzazione delle aspettative EM* (Blei et al., 2003 [34]), *l'inferenza bayesiana variazionale VB* (Blei et al., 2003 [34]), il *campionamento di Gibbs collassato* (Griffiths e Steyvers, [40] 2004) .

Qualunque sia il metodo scelto, lo scopo è sempre il medesimo: approssimare la distribuzione a posteriori ed ottenere una stima dei parametri tramite ottimizzazione. In questa tesi si è scelto di approfondire l'algoritmo di Gibbs Sampling.

6.5 Gibbs Sampling

Il campionamento di Gibbs è un membro della classe di algoritmi del framework Markov chain Monte Carlo, detti MCMC, che è utilizzato in LDA allo scopo di inferire le variabili latenti dei topic nei documenti.

6.5.1 Catena di Markov

Le catene di Markov sono dei sistemi dinamici che descrivono un processo stocastico particolare in cui fenomeni casuali evolvono in funzione del tempo e non hanno memoria degli stati precedenti. Un processo stocastico è definito come una sequenza di variabili aleatorie $\{X_0, X_1, \ldots, X_n, \ldots\}$, dove ogni X_t determina lo stato del sistema al tempo t e assume numero finito di valori. Se la distribuzione di probabilità condizionata di uno stato X_t dipende esclusivamente dal precedente X_{t-1} allora si ha un processo di Markov, che viene definito come un processo privo di memoria in quanto la sua evoluzione al tempo t dipende solamente dal valore che assume il suo stato in quel momento.

		A	B	C	D
	A	0	1	0	0
$T =$	B	0.5	0	0.2	0.3
	C	0	0	0	1
	D	1	0	0	0

Tabella 6.1: Esempio di una matrice di transizione per una catena di Markov.

Una catena di Markov discreta a stati finiti può essere rappresentata facilmente come una matrice o come un grafo. Nello specifico, la rappresentazione per mezzo di una matrice è particolarmente adatta a descrivere la distribuzione di probabilità degli stati a qualsiasi passo t della catena. Una matrice quadrata $P = [p_{ij}]$ di un processo di Markov viene detta matrice di transizione, ed ha come elementi le probabilità che il sistema esegua una transizione dallo stato i allo stato j. Le righe della matrice sommano a 1 poiché rappresentano la probabilità totale di transizione del sistema ad uno degli stati possibili; mentre gli elementi appartenenti alla diagonale esprimono la probabilità che il sistema rimanga nello stato attuale. Se si vuole descrivere le probabilità di transizione dopo N passi, la matrice P deve essere moltiplicata N volte per sè stessa,

ottenendo così la matrice P^N. Aumentando il numero di passi è possibile che le probabilità di transizione rimangano invariate. In questi casi, il sistema raggiunge uno stato stazionario. Se questo non viene mai raggiunto allora è possibile che il sistema ammetta delle periodicità [41].

6.5.2 Descrizione dell'algoritmo

L'obiettivo del Gibbs Sampling è quello di costruire una catena di Markov che abbia la distribuzione a posteriori come distribuzione stazionaria. Come detto, una catena di Markov è una sequenza di variabili aleatorie, dove ognuna dipende dalla precedente. Ogni stato della catena è rappresentato dalle variabili z_{dn}, che rappresentano le assegnazioni degli argomenti per ciascuna parola. Le transizioni tra stati successivi, invece, avvengono tramite il campionamento sequenziale degli assegnamenti z_{dn} da una distribuzione condizionale $P(z_{dn} = k \mid z_{-dn}, w_{dn}, d)$; nella quale z_{dn} corrisponde all'argomento campionato k assegnato alla parola w_{dn}, e z_{-dn} rappresenta le assegnazioni dell'argomento in questione per tutte le altre parole. Le variabili campionate vengono dunque modificate sequenzialmente fino a quando non viene raggiunta la distribuzione stazionaria della catena. Dopo aver stimato la distribuzione a posteriori di z_{dn}, essa viene sfruttata per dedurre le distribuzioni $\phi_{1:T}$ e $\theta_{1:M}$.

In [27] la distribuzione condizionale del Gibbs Sampling è definita come:

$$P(z_{dn} = k \mid z_{-dn}, w_{dn}, d) \propto \frac{C^{VT}_{w_i,k} + \beta}{\sum_{w=1}^{V} C^{VT}_{w,k} + V\beta} \cdot \frac{C^{MT}_{d,k} + \alpha}{\sum_{t=1}^{T} C^{MT}_{d,t} + T\alpha} \tag{6.7}$$

dove:

- C^{MT} è una matrice di dimensione $M \times T$ i cui elementi rappresentano il numero volte che l'argomento k è stato assegnato ad una qualsiasi parola del documento d.
- C^{VT} è una matrice di dimensione $V \times T$ i cui elementi rappresentano il conteggio di quante volte la parola w_i è stata assegnata all'argomento k.
- $\sum_{t=1}^{T} C^{MT}_{d,t}$ è il conteggio totale di argomenti assegnati alle parole nel documento d.
- $\sum_{w=1}^{V} C^{VT}_{w,k}$ è il conteggio totale di parole assegnate all'argomento k.

Nota L'equazione 6.7 fornisce la probabilità non normalizzata. La probabilità effettiva di assegnare un token di una parola ad un argomento si calcola dividendo la quantità definita per il topic k nell'equazione 6.7 per la somma su tutti gli argomenti T .

6.5.3 Esecuzione dell'algoritmo

Avendo ottenuto la distribuzione condizionale completa, l'algoritmo Monte Carlo è di facile esecuzione.

Per ogni documento $\{1, \ldots, d, \ldots, M\}$ del corpus:

1. Si assegna a ciascuna parola w_{dn} del documento un argomento casuale $\{1, \ldots, k, \ldots T\}$.

2. Per ciascuna parola w_{dn}, si iterano i seguenti passaggi:

   (a) Viene tolta l'associazione della parola w_{dn} all'argomento k. Conseguentemente gli elementi $C^{VT}_{w_i,k}$ e $C^{MT}_{d,k}$ vengono decrementati di uno.

   (b) Un nuovo argomento viene campionato dalla distribuzione 6.7 ed assegnato alla parola w_{dn}: le matrici C^{VT} e C^{MT} vengono aggiornate di conseguenza. Nello specifico, il nuovo argomento viene campionato sulla base di:

   - quanto prevale il topic k nel documento d, espresso dal rapporto $\frac{C^{MT}_{d,k}+\alpha}{\sum_{t=1}^{T} C^{MT}_{d,t}+T\alpha}$
   - quanto prevale la parola w_i nel topic k, espresso dal rapporto $\frac{C^{VT}_{w_i,k}+\beta}{\sum_{w=1}^{V} C^{VT}_{w,k}+V\beta}$

Dunque, durante ogni iterazione, si passa attraverso ciascun termine del corpus e si aggiorna l'assegnazione dei topic utilizzando una distribuzione condizionale basata sulle assegnazioni attuali per tutte le altre parole (z_{-dn}).

Al termine di un passaggio completo su tutte le parole, si ottiene un campione di Gibbs che rappresenta lo stato corrente delle associazioni z_{dn} per tutto il corpus di documenti. Questo processo viene ripetuto molte volte: durante le fase iniziale, detta periodo di burn-in, i campioni vengono scartati poiché non rappresentano accuratamente la distribuzione a posteriori, poi i campioni successivi, che iniziano a convergere verso la distribuzione target, vengono salvati a intervalli regolari per ridurre le correlazioni tra campioni consecutivi.

6.5.4 Stima di $\theta_{1:M}$ e $\phi_{1:T}$

Come illustrato, l'algoritmo Gibbs Sampling fornisce stime dirette delle variabili z_{dn}, tuttavia, per valutare il contenuto tematico dei documenti, si richiedono anche le stime di $\theta_{1:M}$ e $\phi_{1:T}$.

Queste grandezze possono essere ottenute facilmente tramite i due rapporti definiti precedentemente dalla distribuzione condizionale 6.7:

$$\theta_{d,k} = \frac{C^{MT}_{d,k} + \alpha}{\sum_{t=1}^{T} C^{MT}_{d,t} + T\alpha} \tag{6.8}$$

$$\phi_{k,w_i} = \frac{C^{VT}_{w_i,k} + \beta}{\sum_{w=1}^{V} C^{VT}_{w,k} + V\beta} \tag{6.9}$$

6.5.5 Esempio

Si riporta un breve esempio preso da una lezione dell'Università di Washington [43] con cui si cerca di spiegare l'algoritmo in una modalità più semplice e intuitiva.

$\mathbf{z_{dn}}$	3	2	1	3	1
$\mathbf{w_{dn}}$	Etruscan	trade	price	tample	market

Tabella 6.2: (1) Inizialmente per ciascuna parola w_{dn} di un documento d viene assegnato un topic $\{1, 2, 3\}$. L'esempio per semplicità riporta un solo documento, ma questa procedura deve essere replicata per ciascun testo del corpus.

	Topic 1	Topic 2	Topic 3
Doc d	2	1	2

Tabella 6.3: Viene costruita la matrice C^{MT} che contiene il numero di volte che l'argomento k è stato assegnato alle parole del documento d.

	Topic 1	Topic 2	Topic 3
Etruscan	1	0	35
Market	50	0	1
price	42	1	0
tample	0	0	20
trade	10	8	1

Tabella 6.4: Viene costruita anche la matrice C^{VT} che contiene il conteggio di quante volte il token della parola w_i è stato assegnato all'argomento k considerando tutto il corpus.

$\mathbf{z_{dn}}$	3	?	1	3	1
$\mathbf{w_{dn}}$	Etruscan	trade	price	tample	market

Tabella 6.5: $(2)(a)$ Viene tolta l'associazione della parola $w_{d2} =$ *trade* per l'argomento 2.

In generale, ci si aspetta che un documento tratti principalmente pochi e specifici temi: per questo motivo, i campionamenti vertono a rendere i documenti d i più affini possibile ad un topic k e ad associare nella misura maggiore le parole w_{dn} ad uno specifico argomento. La probabilità di campionare un topic k, infatti, è influenzata da quanto esso prevale nel documento d, e dal

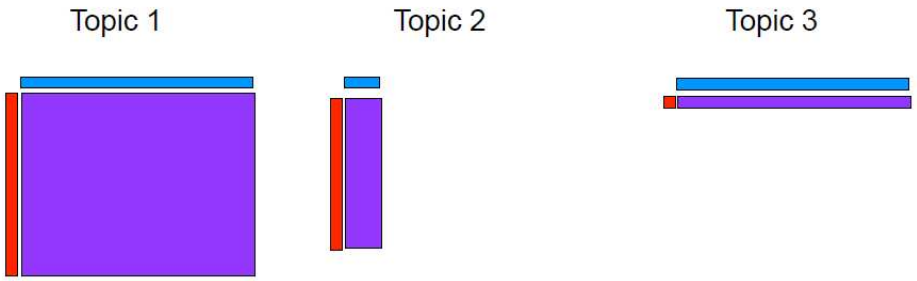

Figura 6.5: Stadio $(2)(b)$ del Gibbs Sampling: viene selezionato un nuovo topic dalla distribuzione $P(z_{dn} = k \mid z_{-dn}, w_{dn}, d)$.

grado con cui è associato alla parola w_i. In questo caso il topic 1 ha più probabilità di essere estratto perché è presente due volte nel documento (come il topic 3, il topic 2 invece è presente una sola volta), ed ha dieci associazioni con la parola *trade* (il topic 3 ne ha solo una e il topic 2 ne ha otto).

$\mathbf{z_{dn}}$	3	**1**	1	3	1
$\mathbf{w_{dn}}$	Etruscan	trade	price	tample	market

Tabella 6.6: Immaginando di aver selezionato il topic 1 si procede aggiornando le assegnazioni z_{dn}.

	Topic 1	Topic 2	Topic 3
Doc d	**3**	**0**	2

Tabella 6.7: Aggiornamento della matrice C^{MT}.

	Topic 1	Topic 2	Topic 3
Etruscan	1	0	35
market	50	0	1
price	42	0	1
tample	0	0	20
trade	**11**	**7**	1

Tabella 6.8: Aggiornamento della matrice C^{VT}. Poi le iterazioni continuano selezionando progressivamente tutte le parole del corpus.

Capitolo 7

Valutazione degli algoritmi di Topic Modeling

Abbiamo visto come i topic models riescano a rappresentare e riassumere il contenuto di grandi collezioni di documenti. Tuttavia, un ostacolo significativo alla loro adozione, è l'estrazione di argomenti di bassa qualità, ossia che combinano concetti non correlati o vagamente correlati. Questo, chiaramente, ha portato ad un crescente interesse per la stima della qualità dei modelli, dove l'obiettivo principale è identificare una grandezza in modo automatico, ossia senza necessità di valutazioni da parte di persone, che rappresenta il grado con cui gli argomenti concordano con i giudizi umani. In altre parole, lo scopo è sviluppare delle misure che permettono di distinguere gli argomenti semanticamente interpretabili rispetto agli argomenti che sono solo artefatti arbitrari di inferenza statistica.

7.1 Topic coherence

Le metriche di coherence per un argomento si basano sul calcolo di punteggi che definiscono il grado con cui i termini si supportano a vicenda. Questo avviene tramite la definizione di un punteggio di "somiglianza" per tutte le coppie di termini di un argomento che, sommati assieme, permettono di ottenere un valore complessivo che definisce la "coerenza" dell'argomento [1]. Questo concetto viene espresso dalla seguente espressione :

$$\text{coherence}(W) = \sum_{(w_i, w_j) \in W} \text{score}(w_i, w_j; \epsilon) \tag{7.1}$$

[1] Esistendo diverse varianti di metriche di coherence, il punteggio complessivo non viene sempre ricavato sulla base di una somma dei punteggi intermedi. In alcuni casi, ad esempio, può essere calcolata la media dei valori, in modo tale che il risultato sia normalizzato e non dipenda dal numero di termini considerati. In generale, si possono sviluppare diversi approcci per rendere la coherence di un topic confrontabile tra modelli che utilizzano argomenti di dimensione differente.

Dove W rappresenta l'insieme dei termini dell'argomento, mentre ϵ è un fattore di smorzamento che può essere utilizzato come regolatore dell'effetto dei termini meno frequenti. Nella pratica, si considerano semplicemente i primi n termini più rilevanti dell'argomento in quanto corrispondenti alle più alte correlazioni con il giudizio umano [21].

Compresa l'idea di base su cui si fonda la coherence dei topic, ora vengono trattate due metriche differenti che, secondo Röder, Both e Hinneburg (2015 [5]) e Mimno et al. (2011 [44]) , hanno una buona correlazione con i giudizi umani sulla qualità degli argomenti: UMass metric e Cv metric.

7.1.1 Metrica UMass (Università del Massachusetts)

La metrica UMass (Mimno et al. 2011 [44]) definisce la coherence sulla base del numero di occorrenze dei termini nei documenti. Questi conteggi si riferiscono al corpus che stiamo osservando e, per questo motivo, viene definita come una metrica intrinseca.

La formula atta al calcolo dei punteggi intermedi, detti misure di conferma, è la seguente:

$$\text{score}(w_j; w_i; \epsilon) = \log\left(\frac{D(w_j; w_i) + \epsilon}{D(w_i)}\right) \tag{7.2}$$

Dove:

- $D(w_j; w_i)$: rappresenta il conteggio del numero di documenti contenenti i termini w_j e w_i.
- $D(w_i)$: è il conteggio del numero di documenti contenenti w_i.
- ϵ: come detto precedentemente, è un valore di smoothing di piccole dimensioni, necessario per includere anche termini meno frequenti ed evitare divisioni per zero.

Ciascun termine appartenente all'insieme $\{w_1, \ldots, w_n\}$, ordinato in base alla probabilità delle rispettive parole di descrivere l'argomento in esame, viene confrontato rispettivamente con i termini precedenti secondo l'equazione 7.2. Poi, la media dei diversi valori ottenuti corrisponde al punteggio di coerenza complessivo per l'argomento. Si noti che i punteggi intermedi sono per la maggior parte negativi poiché si effettua il logaritmo di una quantità minore di uno (il numero di occorrenze di un termine con un altro è minore o uguale al numero di occorrenze del termine stesso). Segue che valori più vicini allo zero implicano parole che tendono a co-occorrere più spesso, denotando maggiore qualità del topic e quindi del modello.

7.1.2 Metrica Cv (Coherence value)

Si tratta di una metrica sviluppata da Roder, Both, e Hinneburg nel 2015 [5] che, come prima fase, crea coppie di termini a partire dall'insieme $W = \{w_1, \ldots, w_n\}$ composto dai primi n termini più importanti di ciascun argomento. Le coppie, nello specifico, si formano nel seguente modo: si seleziona ciascun termine w_i, che formerà il sottoinsieme W' costituito da un solo elemento, e si associa al sottoinsieme W'' , che conterrà la parola w_i più tutti i termini successivi dell'insieme W stesso. Ad esempio, se $W = \{w_1, w_2, w_3\}$, allora una coppia è costituita dal sottoinsieme ad elemento singolo $W' = \{w_1\}$ e dal sottoinsieme $W'' = \{w_1; w_2; w_3\}$.

Si seleziona quindi ogni coppia $S_i = \{W''; W'\}$ su cui viene calcolato un punteggio (misura di conferma) che indica quanto W'' assomigli W', rispetto a tutte la parole di W. Tale somiglianza viene calcolata per mezzo dell'informazione reciproca puntuale normalizzata (NPMI), come mostrato nell'equazione 7.3.

$$\text{NPMI}(w_i, w_j)^\gamma = \left(\frac{\log\left(\frac{p(w_i,w_j)+\epsilon}{p(w_i)\cdot p(w_j)}\right)}{-\log(p(w_i, w_j) + \epsilon)} \right)^\gamma \tag{7.3}$$

Dove:

- $p(w_i, w_j)$: rappresenta la probabilità congiunta che i due termini $w_j \in W'$ e $w_j \in W''$ occorrano assieme nelle stesso documento.
- $p(w_i)$: indica la probabilità che il termine w_i sia presente nel documento.
- $p(w_j)$ indica la probabilità che il termine w_j sia presente nel documento.
- γ serve a dare più peso ai valori NPMI.
- ϵ è il solito valore di smoothing, che in questo caso si include per evitare problemi con i logaritmi di zero.

In particolare, la probabilità congiunta che i due termini w_i e w_j siano presenti assieme viene calcolata attraverso un algoritmo a finestra di scorrimento. Si definisce una finestra di dimensione fissa che scorre attraverso un set di documenti che, per ogni posizione, conta il numero di volte in cui entrambi i termini occorrono assieme. La probabilità congiunta è quindi calcolata come il rapporto di questo conteggio e il numero totale di finestre. Lo stesso ragionamento si ripete per le probabilità $p(w_i)$ e $p(w_j)$ dove, ovviamente, si effettua il semplice conteggio delle singole parole attraverso lo scorrimento della finestra. In questo approccio, il set di documenti può provenire anche da un corpus esterno di riferimento, rendendo così la metrica Cv estrinseca, ovvero che calcola la somiglianza tra termini attraverso un set di

addestramento su cui non è stato definito il modello.

Per avere avere una buona stima del modello in termini di bontà dei topic formulati, si dovrebbero combinare più tecniche di valutazione [21]. Per quanto concerne la coerenza degli argomenti, l'utilizzo di metriche intrinseche ed estrinseche permette di avere una visione a tutto tondo dei risultati, poiché in questo modo riflettono aspetti diversi dell'interpretabilità dei topic: un approccio intrinseco stabilisce come i termini si confermino tra loro rispetto al documento utilizzato per creare gli argomenti, mentre una misura estrinseca, attraverso un corpus esterno, definisce come i termini si confermino a vicenda in un senso più generale.

7.2 Numero di topic

Un aspetto fondamentale da stabilire per l'esecuzione di un modello LDA è il numero di topic T da utilizzare. A priori, non è possibile conoscere quale valore possa essere il più adeguato per una determinata collezione di testi. Si pensi all'esempio in cui viene considerato un numero elevato, per ogni argomento si avranno associati pochi documenti, aumentando così la coerenza ma, allo stesso tempo, si abbasserà la diversità tra i topic stessi. Per contro, avere una configurazione con poche categorie, comporterà l'unione di temi differenti compromettendo la coerenza di quest'ultimi. Questa problematica, ovvero determinare quali aspetti favorire, può essere affrontata confrontando metriche di valutazione dei modelli caratterizzate da un numero T differente. Facendo riferimento alla *topic coherence*, si può sviluppare una funzione che descrive l'andamento della coerenza dei topic al variare di differenti configurazioni, come si osserva in figura 8.1.

7.3 Altre metriche di valutazione dei topic

Per valutare in modo appropriato i topic ottenuti tramite LDA, sono state sviluppate ulteriori metriche diagnostiche. Nello specifico, in questa sezione verranno proposte le misure che fornisce la libreria Mallet per il topic modeling [46].

- *Token*: questa metrica indica il numero di termini assegnati al topic k. Generalmente, valori troppi piccoli o elevati sono indicatori di temi poco affidabili: nel primo caso si dispongono di poche osservazioni per stabilire un'accurata distribuzione delle parole; nel secondo invece è possibile che l'argomento in questione sia costituito in prevalenza da termini da considerare come *stopwords*.

- *Document entropy*: indica la dispersione di un topic tra i documenti. Un topic caratterizzato da una bassa entropia sarà concentrato in pochi documenti, mentre un valore più elevato implica che sarà più uniformemente distribuito.

- *Word-length*: rappresenta la lunghezza media delle parole più importanti per il topic. Alle parole più lunghe si associa un significato più specifico: segue che gli argomenti che riuniscono parole corte probabilmente sono poco settoriali.

- *uniform-dist*: misura la distanza della distribuzione di un argomento sulle parole da una distribuzione uniforme. Si desiderano valori elevati poiché denotano maggiore specificità dei topic.

- *Exclusivity*: è una metrica che definisce quanto le parole principali per un argomento siano esclusive per lo stesso. Se tale valore è elevato significa che le parole non concorrono per definire anche altri argomenti.

- *rank 1 documents*: permette di distinguere gli argomenti ”reali” da quelli che emergono a causa del contesto nel quale i documenti sono inseriti. Ad esempio, ci saranno dei casi in cui alcuni documenti approfondiscono specificatamente dei temi, come la musica o lo sport, rendendo evidente il topic principale. Altri, invece, includeranno molti termini e frasi ricorrenti, che non contribuiscono direttamente allo sviluppo del tema centrale del documento: fanno parte solamente del gergo comune in quel contesto. In generale, se un argomento è abbastanza raro nei documenti e quando compare è associato ad un elevato numero di parole, allora probabilmente si tratta di un topic informativo. Al contrario, un tema che è spesso ricorrente e distribuito in tanti documenti senza mai dominare il discorso, sarà solo un argomento di sfondo.

Queste misure di tipo statistico possono essere affiancate anche da semplici valutazioni preliminari qualitative dei risultati. In questo approccio i topic vengono distinti in categorie [47]:

- Topic distinti da parole troppo generiche o specifiche.

- Topic misti o collegati: ovvero con sottoinsiemi di parole incoerenti o sottoinsiemi di parole che condividono più interpretazioni.

- Topic con parole identiche a quelle di un altro topic.

- Topic costituiti da stopwords.

- Topic con parole casuali o non interpretabili.

Bisogna sottolineare che lo scopo è sviluppare degli argomenti semanticamente interpretabili e comprensibili a livello individuale. La valutazione del modello quindi verte sempre alla verifica che quest'ultimo sia effettivamente in grado di produrre ulteriore conoscenza sul materiale in esame.

Capitolo 8

Esempio di Applicazione dell'Algoritmo LDA

8.1 Librerie

Con la definizione dei topic models sono state sviluppate diverse librerie dedicate che hanno permesso di rendere queste tecniche più accessibili e scalabili. Le librerie più comuni sono:

- *Gensim* [48]: si tratta di una libreria Python che permette di utilizzare diversi algoritmi di topic modeling, inclusi LSA e LDA.
- *Mallet* [45]: è una libreria Java per il natural language processing che include tra le sue funzionalità la classificazione e clustering di documenti, il topic modeling e altre applicazioni per il text mining.
- *Stanford topic modeling toolbook (TMT)* [49]: è un framework scritto in linguaggio Scala sviluppato per aiutare i ricercatori ad analizzare grandi volumi di dati, compresi file Excel o altri fogli di calcolo. Tra i diversi algoritmi inclusi ci sono LDA, Labeled LDA e PLDA.
- *Pacchetti open source* [50]: il gruppo di ricerca di David Blei (autore di LDA) ha sviluppato diversi pacchetti open source gratuiti in c, c++ e Python rilasciati su GitHub. Tra le varie implementazioni incluse nei pacchetti, troviamo algoritmi come: l'EM variazionale per LDA, l'inferenza variazionale per modelli di argomento collaborativi, processi di Dirichlet gerarchici (HDP), l'allocazione di Dirichlet latente gerarchica (hLDA), LDA supervisionata (sLDA), e topic models correlati (CTM).

In questo esempio viene utilizzato un wrapper Mallet per Python sviluppato da Maria Antoniak [51], che permette di integrare le diverse funzionalità offerte da Mallet con gli strumenti, librerie e framework dell'ambiente Python.

8.2 Dataset

Il dataset soggetto alle analisi è un sample casuale di 1051 articoli estratti dall'insieme di tutte le pubblicazioni uscite da gennaio 2010 a dicembre 2024 su testate giornalistiche in lingua italiana, quali: Avvenire, Corriere della Sera, Giornale, La Stampa, Mattino, Messaggero, La Repubblica e Il Sole 24 Ore.

I documenti sono stati selezionati ed indicizzati da un sample di 2786124 articoli tramite una piattaforma chiamata TIPS [52]–[54] utilizzando la query:

```
\"intelligenza artificiale\" OR
\"artificial intelligence\" OR
\"machine learning\" OR
\"apprendimento automatico\" OR
\"Large Language Model\" OR
\"Large Language Models\" OR
\"LLM\" OR
\"chatGPT\"
```

in cui le espressioni tra virgolette indicano che le parole devono apparire una accanto all'altra nei documenti considerati più rilevanti dalla query: la stringa "*intelligenza artificiale*" deve apparire esattamente così nel testo (a meno di trasformazione in lowercase).

Dunque, i documenti scaricati vengono raccolti in un unico file "di training" in formato *txt*. Nello specifico, per ogni documento viene estratto il nome (ID del documento), la data e il testo, che saranno inseriti in una riga del file separati da uno spazio di tabulazione. Questa operazione viene realizzata semplicemente tramite Python utilizzando le librerie *glob* e *Path*.

8.3 Elaborazione preliminare e trasformazione dei dati

I documenti vengono segmentati in token, ovvero vengono scissi in unità minime che possono variare in base alle necessità delle analisi (e.g. la lingua dei documenti). Mallet, in modo predefinito, utilizza un'espressione regolare che, supportando l'utilizzo di tutte le lettere Unicode, dei trattini, degli apostrofi e degli acronimi con notazione puntata, si adatta molto bene alla gestione dei testi in inglese. Nel caso si avessero esigenze particolari di tokenizzazione è possibile specificare un nuovo criterio di suddivisione delle parole utilizzando l'opzione di Mallet *–token-regex*.

Vengono ora riportate le operazioni di "pulizia" che mirano ad ottenere un dataset coerente e privo di "rumore" per non compromettere le analisi successive.

Normalizzazione In questa fase le lettere maiuscole vengono convertite in minuscolo al fine di rendere uniformi i token estratti ed eliminare possibili duplicazioni. In modo predefinito, Mallet normalizza i caratteri in minuscolo, tuttavia, nel caso in cui si volesse preservare lo stato originale delle parole, si può semplicemente specificare il comando *–preserve-case*

Nonostante non sia utilizzata in questo esempio, viene riportata anche un'altra tecnica di normalizzazione dei dati: lo stemming. Si tratta di un metodo che permette di trasformare le parole nella loro forma flessa e più generale, riconducendole alla loro radice (stem). Dunque, vengono rimossi tutti gli affissi dei termini formando un elemento linguistico irriducibile che continua a esprimere il significato del termine completo [55].

Gli algoritmi di stemming, tuttavia, utilizzando regole euristiche che permettono semplicemente di troncare i vari termini, possono produrre delle parole non valide. Per questo motivo spesso tale tecnica può essere sostituita con la lemmatizzazione, ovvero una variante più complessa e sofisticata che utilizza analisi morfologiche e fa riferimento alle forme delle parole nel dizionario per garantire maggiore correttezza dei risultati [56].

Stopwords Tale operazione prevede l'eliminazione di una serie di parole, quali avverbi, congiunzioni, pronomi e preposizioni molto comuni e di uso regolare che, essendo prive di significato semantico di rilievo, possono costituire rumore per l'analisi e rivelazione dei topic. Con Mallet si può eseguire questa cernita con il comando *–remove-stopwords*

La libreria prevede anche l'utilizzo dell'opzione *–keep-sequence*, che permette di preservare l'ordine logico delle parole durante il processo di trasformazione del testo nel formato adatto all'analisi. Quindi, utilizzando questo flag, i termini non verranno semplicemente conteggiati, ma verranno considerati anche in relazione a come si susseguono nei documenti.

```
os.system(path_to_mallet  + ' import-file --input "'
                + path_to_training_data + '"'
                + ' --output "' + path_to_formatted_training_data + '"' \
                + ' --keep-sequence' \
                + ' --use-pipe-from "' + use_pipe_from + '"'
                + ' --preserve-case')
```

8.4 Estrazione dei topic

In questa fase vengono definiti i comandi Mallet per addestrare il modello LDA deputato all'identificazione dei temi latenti. Ogni parametro utilizzato permette di specificare differenti

esecuzioni e output, che possono variare a seconda degli scopi e delle necessità delle analisi.

Un aspetto di fondamentale importanza per il modello è la scelta del numero di topic che bisogna derivare dai documenti (*–num-topics*). Questo valore, in generale, può variare e seguire le nostre preferenze, nonostante ciò, come descritto nella sezione 7.2, può essere ottimizzato confrontando metriche specifiche, tra cui la coherence. Dunque, sono state svolte più esecuzioni del modello per tracciarne l'andamento al variare del numero di argomenti. Avendo selezionato un sample abbastanza semplice, quello che si nota (figura 8.1) è che la coerenza dei topic tendenzialmente diminuisce all'aumentarne del numero. Sono stati scartati, tuttavia, valori troppo piccoli poiché non fornirebbero abbastanza dettaglio dei temi specifici trattati dal corpus. La scelta finale, presa dopo una breve analisi qualitativa dei risultati, è stata il numero 12 a cui corrisponde un massimo locale nel grafico, oltre che argomenti ben definiti per la maggior parte.

Figura 8.1: Coherence UMass in relazione al numero di topic

Un altro elemento da stabilire è il numero di parole importanti che definiscono un topic. Il comando *–num-top-words* permette di mettere in relazione ogni argomento con un determinato numero di parole che, ordinate in termini di rilevanza, ci permettono di distinguere e etichettare i vari argomenti. Queste associazioni saranno disponibili nel file *-output-topic-keys* che, se esaminato, ci permetterà di attribuire agli argomenti un nome piuttosto che un numero.

In questo stadio vengono definiti anche gli output del processo. Tra i più importanti distinguiamo quelli prodotti da *–output-doc-topics* e *–topic-word-weights-file* che corrispondono rispettivamente alle distribuzioni dei topic nei documenti (θ_d) e ai pesi associati alle parole per ciascun argomento (da cui si può ricavare ϕ_z). Altri file di output sono:

- *–output-state*: un file che salva lo stato del modello durante l'addestramento, indicando per ciascuna parola del vocabolario il topic di riferimento.
- *–diagnostics-file*: contiene le valutazioni riferite ai singoli topic (sezione 7.3).
- *–output-topic-docs*: per ogni topic descrive i documenti più affini in ordine decrescente.

```
os.system(path_to_mallet + ' train-topics --input "'
        + path_to_formatted_training_data + '"' \
        + ' --num-topics ' + str(num_topics) \
        + ' --random-seed 17 ' \
        + ' --num-top-words 100 ' \
        + ' --inferencer-filename "' + path_to_model + '"' \
        + ' --output-topic-keys "' + path_to_topic_keys + '"' \
        + ' --output-doc-topics "' + path_to_doc_topic_proportions + '"' \
        + ' --output-model "' + path_to_output_model + '"' \
        + ' --output-state "' + path_to_output_state + '"' \
        + ' --output-topic-docs "' + path_to_top_topic_docs + '"' \
        + ' --num-top-docs 1000000' \
        + ' --topic-word-weights-file "' + path_to_word_weights + '"' \
        + ' --diagnostics-file "' + path_to_diagnostics + '"' \
        + ' --optimize-interval 10')
```

8.5 Risultati

Per primo, vengono visionati i file prodotti dalla direttiva *–output-topic-keys* che, come detto, ci permettono di stabilire quali argomenti sono trattati nei testi. Quindi i topic vengono individuati tramite la visualizzazione dei principali termini che li compongono: si possono usare delle semplice tabelle, oppure delle tecniche di rappresentazione più raffinate, come le *words cloud*.

Figura 8.2: Words Cloud

Topic	Parole									
0	pazienti	covid	medicina	paziente	medici	sanità	diagnosi	euro	salute	medico
1	robot	macchine	umani	umano	macchina	chatgpt	cervello	linguaggio	scienza	umana
2	musica	film	artificiale	festival	serie	nuovo	museo	intelligenza	cinema	cultura
3	miliardi	dollari	milioni	amazon	società	microsoft	mercato	apple	chatgpt	google
4	smartphone	foto	google	app	pixel	samsung	casa	android	versione	permette
5	digitale	lavoro	italia	aziende	imprese	settore	competenze	sviluppo	nuove	tecnologie
6	facebook	social	contenuti	utenti	network	video	online	tiktok	twitter	media
7	cina	stati	governo	guerra	presidente	cinese	paesi	uniti	politica	stato
8	guida	autonoma	auto	sistema	veicoli	nuova	veicolo	bordo	tecnologia	mobilità
9	anni	quando	solo	essere	fare	era	fatto	cosa	poi	stato
10	startup	euro	milioni	chiuso	round	piattaforma	capital	venture	mila	tech
11	artificiale	dati	essere	tecnologia	grado	può	intelligenza	modo	ricerca	solo

Figura 8.3: Parole più importanti per ciascun topic.

Gli argomenti[1] interpretati sono i seguenti:

0. Medicina.
1. Robotica.
2. Produzione artificiale di musica e film.
3. Organizzazioni e costi per AI.
4. Dispositivi mobili e brand tech.
5. Sviluppo e tecnologie per smart working.
6. Social network.
7. Politica.
8. Sistemi di guida autonoma.
9. Argomento non definito: contesto.
10. Startup innovative.
11. Argomento poco specifico: intelligenza artificiale.

In particolare, il corpus nella sua complessità tratta questi topic secondo la distribuzione espressa dalla figura 8.4.

Tramite il file *output-doc-topics*, che di fatto rappresenta i parametri θ_d, è possibile conoscere le percentuali di correlazione dei topic per i documenti. Generalmente i grafici

[1] A differenza dalla trattazione precedente, dove topic sono stati definiti come $\in \{1, \ldots, k, \ldots, T\}$ ed i documenti $\in \{1, \ldots, j, \ldots, M\}$, Mallet inizia la numerazione da 0.

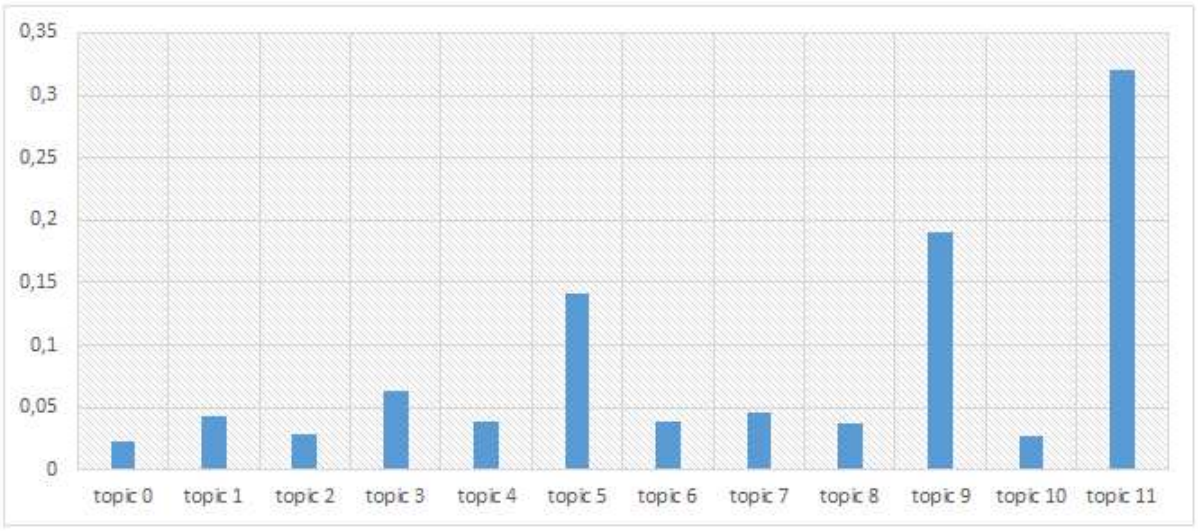

Figura 8.4: Distribuzione dei topic nel corpus

evidenziano il fatto che i testi hanno un topic principale. Questo è in linea con le assunzioni a priori che prevedevano iperparametri $\vec{\alpha}$ minori di 1 per ottenere distribuzioni θ sparse. Facendo riferimento alla figura 8.5 possiamo notare che il documento numero 0 affronta maggiormente i topic 5, 9 e 11 che, sulla base delle nostre interpretazioni, corrispondono al tema dello sviluppo e smart working, oltre che dell'intelligenza artificiale. Come prova, è stato letto il documento in questione che parla del futuro imprenditoriale di Torino e del Piemonte: vengono esplorati diversi ambiti prospettando un avvenire orientato verso l'innovazione soprattutto per settori quali l'agrifood e l'intelligenza artificiale.

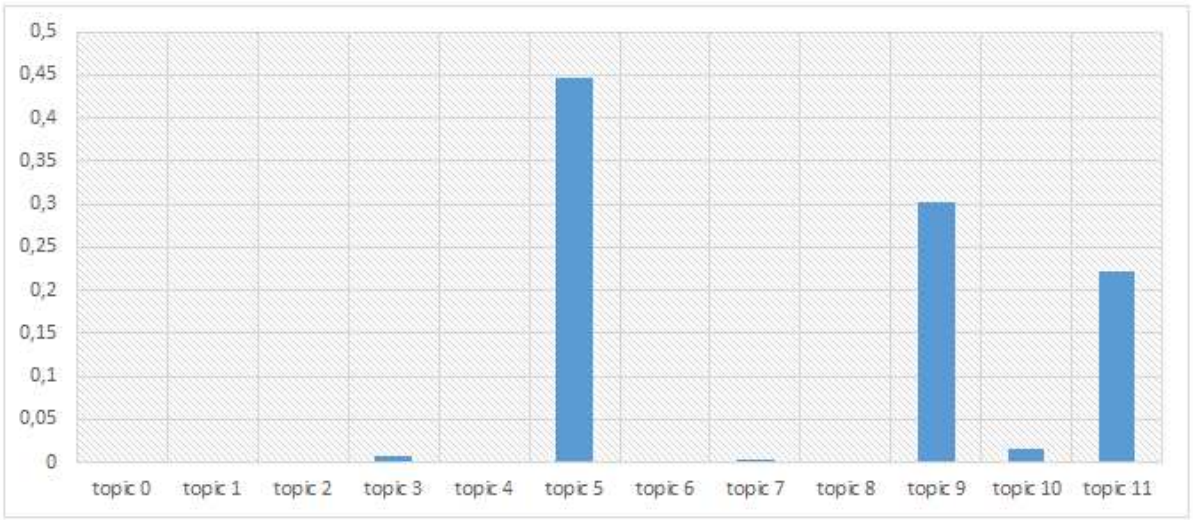

Figura 8.5: Analisi dei topics per il documento 0.

Nel caso in cui un argomento corrispondesse ad un nostro interesse si può consultare il file *output-topic-docs* che descrive per ogni topic i documenti più affini in ordine decrescente. Nel nostro esempio i risultati sono descritti dalla tabella 8.1.

Topic	**Doc**
0	193
1	971
2	585
3	105
4	160
5	338
6	474
7	182
8	97
9	790
10	209
11	382

Tabella 8.1: Documenti più affini per ciascun topic.

Se si desidera conoscere l'evoluzione dei temi affrontati dalle testate giornalistiche nel corso del tempo è possibile procedere con le seguenti analisi. Innanzitutto, viene espresso il numero di documenti disponibili nel corpus in funzione del tempo (8.6). Per ogni anno vengono quindi conteggiati i topic più ricorrenti: si tiene conto del topic principale di ciascun testo (è stato escluso il topic 9 poiché troppo vago). Queste informazioni, presenti nella figura 8.7, possono essere accoppiate con il grafico 8.8, così da vedere i temi affrontati ogni anno in percentuale. Come prevedibile, si nota che il topic prevalente è l'undicesimo, ovvero l'intelligenza artificiale. Inoltre, in maggiore dettaglio si ottiene che:

- Gli argomenti affrontati nel 2010 sono la robotica e la produzione artificiale di musica e film.
- Nel 2013 sono stati pubblicati principalmente articoli che trattano di sistemi di guida autonoma e di robotica.
- Nel 2014 si è parlato di robotica e costi per AI.
- Nel 2015 prevale il tema della produzione artistica artificiale.
- Nel 2016 sono stati affrontati diversi argomenti: sviluppo e tecnologie per smart working, dispositivi mobili e brand tech, sistemi di guida autonoma e organizzazioni e costi per AI.
- Nel 2017 sono stati pubblicati articoli di politica mondiale e sistemi di guida autonoma.

- Nel 2018 gli argomenti più importanti sono lo sviluppo e i sistemi di guida autonoma.
- Nel 2019 si è parlato sia di sviluppo ma anche di produzione artistica con l'intelligenza artificiale.
- Nel 2022, 2021, 2020 vengono trattati principalmente temi inerenti allo sviluppo e smart working.
- Nel 2023 i documenti affrontano argomenti che riguardano: politica, costi per AI, produzione artificiale di musica, sviluppo e smart working.

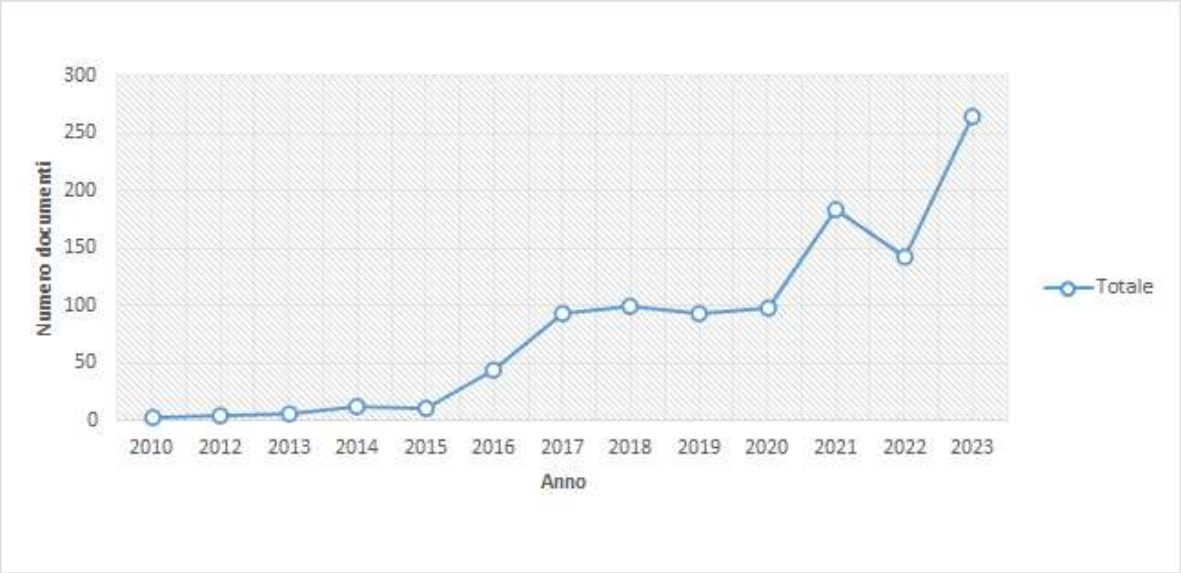

Figura 8.6: Conteggio dei documenti nel corso del tempo

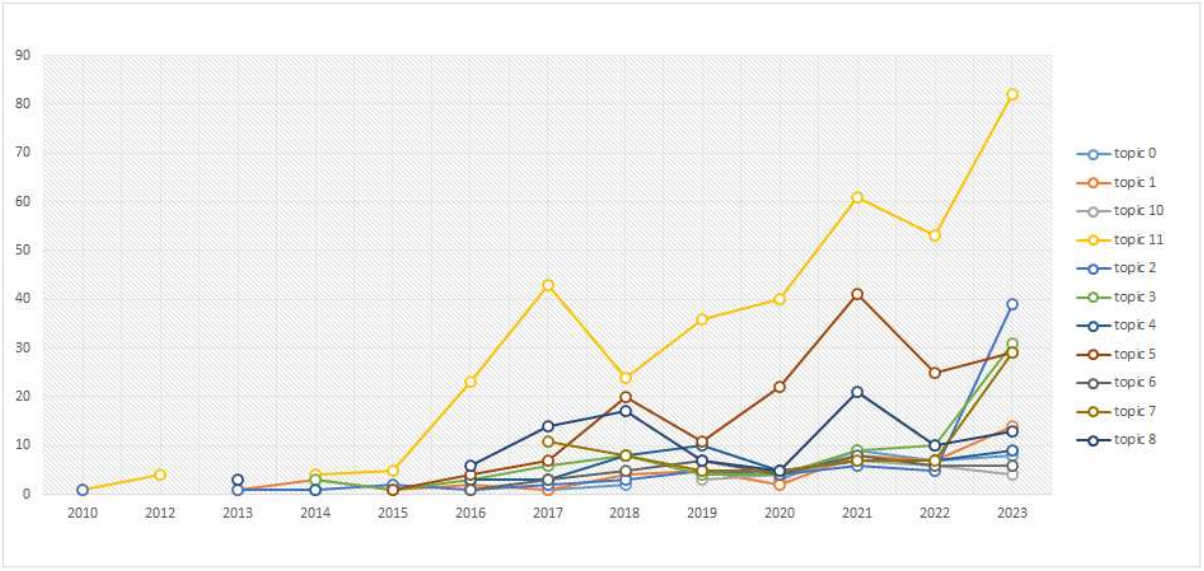

Figura 8.7: Conteggio del numero di documenti appartenenti a ciascun topic

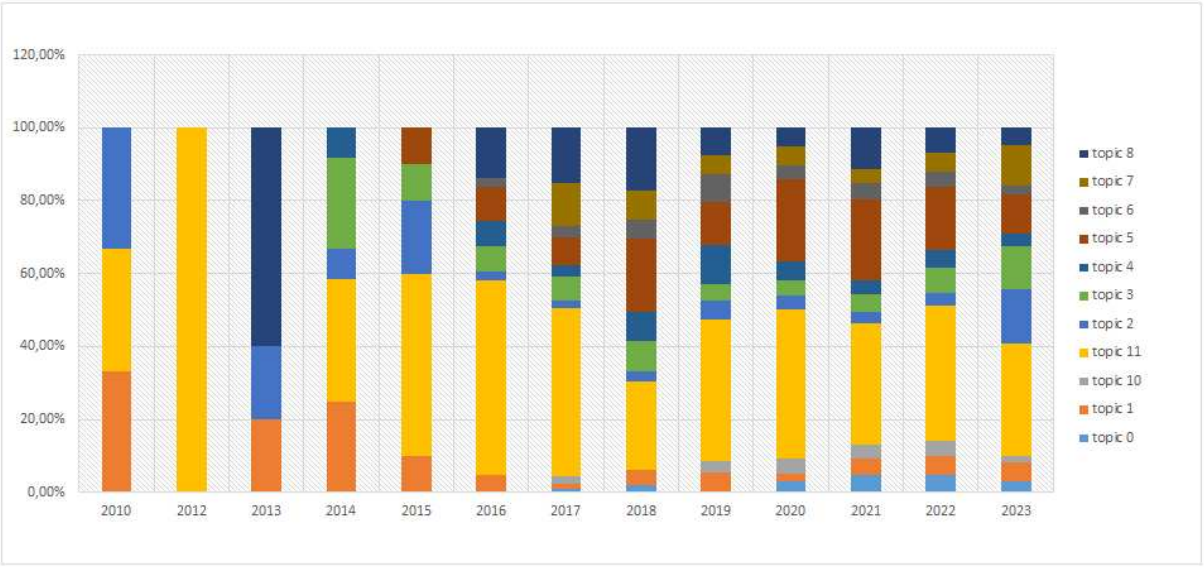

Figura 8.8: Percentuale dei topic trattati negli anni

Capitolo 9

Conclusioni

Nel corso di questa tesi è stato esplorato il tema del topic modeling, analizzando diverse metodologie che rappresentano dei mezzi efficaci per identificare e catalogare gli argomenti principali sviluppati da un insieme di documenti. L'obiettivo principale è stato quello di approfondire questi algoritmi con l'intento di apprezzarne il funzionamento, l'importanza e le potenzialità in contesti applicativi concreti. Abbiamo visto che questi modelli si prestano a diverse applicazioni e, come esempi, si è voluto approfondire brevemente due settori: la giurisprudenza e la bioinformatica. Nel campo della giurisprudenza i topic models vengono utilizzati specialmente al fine di facilitare la ricerca di documenti inerenti a specifiche fattispecie, consentendo indagini più efficienti per la consultazione di grandi archivi legali, oltre a permettere considerazioni comparative tra diverse giurisdizioni e periodi storici. Per quanto riguarda la bioinformatica, queste tecniche sono state adattate e rielaborate per lo studio e la classificazione di complessi dati biologici, rivelandosi molto utili nell'ambito della ricerca biomedica dove possono coadiuvare le analisi.

Sono stati quindi affrontati i metodi più tradizionali che hanno definito le basi per il topic modeling: la Latent Semantic Analysis, la Non-Negative Matrix Factorization, la Probabilistic Latent Semantic Analysis e la Latent Dirichlet Allocation. L'esposizione di questi algoritmi nella tesi ha portato diverse sfide, tra cui la necessità di formulare una notazione unificata che potesse rappresentare correttamente i diversi modelli e le fonti su cui si è basato lo studio. Un'attenzione particolare è stata data agli approcci probabilistici, che hanno permesso di approfondire differenti aspetti statistici di fondamentale importanza per l'ambito dell'intelligenza artificiale: spiccano il processo bayesiano e l'inferenza, ma è stata trattata anche l'importanza di determinare le distribuzioni più appropriate per modellare adeguatamente le variabili. Oltre agli aspetti positivi, sono state evidenziate anche le problematicità, come l'estrazione di temi vaghi costituiti da parole poco correlate. Per valutare la bontà dei risultati, vengono quindi utilizzate delle metriche statistiche, che però spesso devono essere affiancate ad una attività umana per verificarne

la concreta qualità e veridicità.

Tra i modelli trattati, l'LDA si è rilevato il più rappresentativo ed efficace in diversi contesti applicativi e, per questo motivo, nel capitolo finale è stato testato su un campione di articoli provenienti da testate giornalistiche italiane che riguardavano l'intelligenza artificiale. Le analisi hanno fornito una rappresentazione di dodici temi latenti che trattavano diversi ambiti, come la robotica, i sistemi di guida autonoma, la politica e lo sviluppo nel mondo del lavoro. Questo esempio ha mostrato come il topic modeling possa essere utilizzato per comprendere in modo veloce ed automatico le tematiche affrontate dai media, fornendo una panoramica di quali sono gli interessi più rilevanti in uno specifico settore. Nonostante si trattasse di un campione semplice, per ottenere analisi ottimali è stato comunque necessario confrontare diverse esecuzioni e sottoporre i risultati a verifiche e validazioni per accertarne l'attendibilità. Un punto di forza dei topic models invece è la loro scalabilità, ovvero la capacità di poter gestire anche dataset di grandi dimensioni senza inficiare la qualità delle analisi. Ad ogni modo, negli ultimi anni sono stati sviluppati modelli dinamici e tecniche che si basano su reti neurali, come il Neural Topic Modeling, che permettono analisi più precise e flessibili, oltre a consentire ulteriori miglioramenti pure in termini di scalabilità.

In conclusione, possiamo definire i topic models come degli strumenti potenti che, già nei primi anni 2000, rappresentavano l'avanguardia per la comprensione di grandi volumi di dati testuali. Il loro utilizzo continuerà a progredire con lo sviluppo di nuovi algoritmi sempre più sofisticati, che ne perfezioneranno i limiti e ne espanderanno significativamente i campi di applicazione, con un impatto significativo in numerosi settori che permeano la nostra società.

Bibliografia

[1] U. Fayyad, G. Piatetsky-Shapiro e P. Smyth, «From Data Mining to Knowledge Discovery in Databases,» *AI Magazine*, vol. 17, n. 3, p. 37, 1996. doi: 10.1609/aimag.v17i3.1230. indirizzo: https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1230.

[2] E. Treccani. «Sentiment Analysis.» (), indirizzo: https://www.treccani.it/enciclopedia/sentiment-analysis_(altro)/.

[3] F. M. Grifeo, *Ufficio del processo, 100mila procedimenti civili in più ogni anno*. indirizzo: https://ntplusdiritto.ilsole24ore.com/art/ufficio-processo-100mila-procedimenti-civili-piu-ogni-anno-AFdAN6nD.

[4] D. C. G. Dott.ssa Martina Saletta, *Topic Modeling per testi legali*. indirizzo: https://www.giustizia.it/cmsresources/cms/documents/2uni4just_unitri_supdig_topic_modeling.pdf.

[5] M. Röder, A. Both e A. Hinneburg, «Exploring the Space of Topic Coherence Measures,» in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, ser. WSDM '15, Shanghai, China: Association for Computing Machinery, 2015, pp. 399–408, isbn: 9781450333177. doi: 10.1145/2684822.2685324. indirizzo: https://doi.org/10.1145/2684822.2685324.

[6] P. J. Rousseeuw, «Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,» *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987, issn: 0377-0427. doi: https://doi.org/10.1016/0377-0427(87)90125-7. indirizzo: https://www.sciencedirect.com/science/article/pii/0377042787901257.

[7] C. D. Manning, P. Raghavan e H. Schütze, «Scoring, term weighting, and the vector space model,» in *Introduction to Information Retrieval*. Cambridge University Press, 2008, pp. 100–123.

[8] L. Liu, L. Tang, W. Dong, X. Wang e S. Yao, «An overview of topic modeling and its current applications in bioinformatics,» *SpringerPlus*, vol. 5, n. 1, p. 1608, 2016. doi: 10.1186/s40064-016-3252-8.

[9] S. Scalabrin. «DNA microarray.» (2024), indirizzo: https://www.microbiologiaitalia.it/didattica/dna-microarray/#:~:text=Il%20DNA%20microarray%20%C3%A8%20una,se%20tali%20sequenze%20sono%20complementari!&text=Un%20DNA%20microarray%20consiste%20in,sonde)%20attaccate%20al%20supporto%20solido..

[10] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer e R. Harshman, «Indexing by latent semantic analysis,» *Journal of the American Society for Information Science*, vol. 41, n. 6, pp. 391–407, 1990. doi: https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

[11] «Unitn: Department of information engineering and computer science. Term Frequency and Inverted Document Frequency.» (), indirizzo: https://disi.unitn.it/~bernardi/Courses/DL/Slides_11_12/measures.pdf.

[12] D. J. Aldous, «Exchangeability and related topics,» in *École d'Été de Probabilités de Saint-Flour XIII — 1983*, P. L. Hennequin, cur., Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, pp. 1–198, isbn: 978-3-540-39316-0.

[13] MathWorks. «Bag-of-Words.» (), indirizzo: https://it.mathworks.com/discovery/bag-of-words.html.

[14] T. Mikolov, K. Chen, G. Corrado e J. Dean, «Efficient Estimation of Word Representations in Vector Space,» *Proceedings of Workshop at ICLR*, vol. 2013, gen. 2013.

[15] J. Devlin, M.-W. Chang, K. Lee e K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, 2019. arXiv: 1810.04805 [cs.CL]. indirizzo: https://arxiv.org/abs/1810.04805.

[16] «Scomposizione ai valori singolar.» (), indirizzo: https://people.dmi.unipr.it/marino.belloni/Didattica/Archivio/aa2008-09/IngCNAaa2008-09/SVD-pseudoinversa.pdf.

[17] «Chapter 7 The Singular Value Decomposition (SVD).» (), indirizzo: https://math.mit.edu/classes/18.095/2016IAP/lec2/SVD_Notes.pdf.

[18] B. P. C. «Topic Modeling Tutorial – How to Use SVD and NMF in Python.» (), indirizzo: https://www.freecodecamp.org/news/advanced-topic-modeling-how-to-use-svd-nmf-in-python/.

[19] J. Xu. «Topic Modeling with LSA, PLSA, LDA & lda2Vec.» (2018), indirizzo: https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05.

[20] D. Lee e H. Seung, «Learning the parts of objects by non-negative matrix factorization,» *Nature*, vol. 401, pp. 788–791, 1999. doi: 10.1038/44565.

[21] K. Svensson e J. Blad, *Exploring NMF and LDA Topic Models of Swedish News Articles*, 2020.

[22] «Problema di programmazione convessa.» (), indirizzo: http://www.oil.di.univaq.it/didattica/corsi/ro1/Lez3.pdf.

[23] L. Camanzi, «Fattorizzazione Matriciale Non Negativa: algoritmi e applicazioni,» tesi di dott. indirizzo: http://amslaurea.unibo.it/19242/.

[24] Y. Koren, R. Bell e C. Volinsky, «Matrix Factorization Techniques for Recommender Systems,» *Computer*, vol. 42, n. 8, pp. 30–37, 2009, issn: 0018-9162. doi: 10.1109/MC.2009.263. indirizzo: https://doi.org/10.1109/MC.2009.263.

[25] H. Kim, H. Park e L. Eldén, «Non-negative Tensor Factorization Based on Alternating Large-scale Non-negativity-constrained Least Squares,» nov. 2007, pp. 1147 –1151. doi: 10.1109/BIBE.2007.4375705.

[26] T. Hofmann, «Probabilistic latent semantic indexing,» in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '99, Berkeley, California, USA: Association for Computing Machinery, 1999, pp. 50–57, isbn: 1581130961. doi: 10.1145/312624.312649. indirizzo: https://doi.org/10.1145/312624.312649.

[27] M. Steyvers e T. Griffiths, «Probabilistic topic models,» in *Handbook of latent semantic analysis*, Psychology Press, 2007, pp. 439–460.

[28] C. Tufts, *The little book of LDA*. indirizzo: https://miningthedetails.com/LDA_Inference_Book/index.html.

[29] L. Hong, «A tutorial on probabilistic latent semantic analysis,» *arXiv preprint arXiv:1212.3900*, 2012.

[30] «ds4psy_2023.» (), indirizzo: https://ccaudek.github.io/ds4psy_2023/intro.html#license-for-this-book.

[31] «ds4psy_2023.» (), indirizzo: https://docenti-deps.unisi.it/wp-content/uploads/sites/35/2020/05/Lezione-22-Massima-verosimiglianza.pdf.

[32] T. Hofmann, «Hofmann, T.: Unsupervised Learning by Probabilistic Latent Semantic Analysis. Machine Learning 42(1-2), 177-196,» *Machine Learning*, vol. 42, pp. 177–196, gen. 2001. doi: 10.1023/A:1007617005950.

[33] D. Tian, «Research on PLSA model based semantic image analysis: A systematic review,» *Journal of Information Hiding and Multimedia Signal Processing*, vol. 9, pp. 1099–1113, set. 2018.

[34] D. M. Blei, A. Y. Ng e M. I. Jordan, «Latent dirichlet allocation,» *Journal of machine Learning research*, vol. 3, n. Jan, pp. 993–1022, 2003.

[35] «Inferenza Bayesiana di una Proporzione Binomiale: L'Approccio Analitico.» (), indirizzo: https://datatrading.info/inferenza-bayesiana-di-una-proporzione-binomiale-lapproccio-analitico/#:~:text=Distribuzione%20Beta,-In%20questo%20caso&text=Essenzialmente%2C%20quando%20%CE%B1%20diventa%20pi%C3%B9,avere%20pi%C3%B9%20%E2%80%9Ccroci%E2%80%9D)%20.

[36] S. Liu, *The Dirichlet Distribution: What Is It and Why Is It Useful?* 2023. indirizzo: https://builtin.com/data-science/dirichlet-distribution.

[37] N. Cao, *Modelli Latent Dirichlet Allocation ed applicazioni in psicologia*, 2019/2020. indirizzo: https://lilia.dpss.psy.unipd.it/~antonio.calcagni/bin/ths/cao_n1.pdf.

[38] J. Boyd-Graber, Y. Hu, D. Mimno et al., «Applications of topic models,» *Foundations and Trends® in Information Retrieval*, vol. 11, n. 2-3, pp. 143–296, 2017.

[39] G. Toto, *Un algoritmo di topic modeling per microblog*, 2021/2022. indirizzo: https://hdl.handle.net/20.500.12608/11379.

[40] T. L. Griffiths e M. Steyvers, «Finding scientific topics,» *Proceedings of the National Academy of Sciences*, vol. 101, n. suppl 1, pp. 5228–5235, 2004. doi: 10.1073/pnas.0307752101.

[41] P. Dotti. «Le catene di Markov, cosa sono, come si collocano nella Data Science.» (), indirizzo: https://www.ai4business.it/intelligenza-artificiale/la-catena-di-markov-cose-come-si-colloca-nella-data-science/.

[42] W. M. Darling, «A theoretical and practical implementation tutorial on topic modeling and gibbs sampling,» in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 2011, pp. 642–647.

[43] E. Fox, «LDA Collapsed Gibbs Sampler, Variational Inference,» CSE547/STAT548, University of Washington, rapp. tecn., 2015. indirizzo: https://courses.cs.washington.edu/courses/cse547/15sp/slides/LDAsampling-variational-annotated.pdf.

[44] D. Mimno, H. Wallach, E. Talley, M. Leenders e A. McCallum, «Optimizing Semantic Coherence in Topic Models,» in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, R. Barzilay e M. Johnson, cur., Edinburgh, Scotland, UK.: Association for Computational Linguistics, lug. 2011, pp. 262–272. indirizzo: https://aclanthology.org/D11-1024.

[45] A. K. McCallum, *MALLET*, http://mallet.cs.umass.edu/, 2002.

[46] A. K. McCallum, *MALLET*, https://mallet.cs.umass.edu/diagnostics.php, 2002.

[47] J. Boyd-Graber, D. Mimno e D. Newman, «Care and Feeding of Topic Models: Problems, Diagnostics, and Improvements,» in *Handbook of Mixed Membership Models and Their Applications* (CRC Handbooks of Modern Statistical Methods), E. M. Airoldi, D. Blei, E. A. Erosheva e S. E. Fienberg, cur., CRC Handbooks of Modern Statistical Methods. Boca Raton, Florida: CRC Press, 2014.

[48] R. R, *Gensim*, http://radimrehurek.com/gensim/, 2008.

[49] S. University, *Stanford Topic Modeling Toolbox (TMT)*, https://nlp.stanford.edu/software/tmt/tmt-0.4/, 2009-2010.

[50] D. Blei, *Blei Lab*. indirizzo: https://github.com/Blei-Lab.

[51] M. Antoniak, *little-mallet-wrapper*, https://github.com/maria-antoniak/little-mallet-wrapper?tab=readme-ov-file, 2021.

[52] «TIPS Project: Technoscientific Issues in the Public Sphere.» (), indirizzo: https://www.tipsproject.eu/tips/#/public/home.

[53] E. D. Buccio, A. Cammozzo, F. Neresini e A. Zanatta, «TIPS: Search and Analytics for Social Science Research,» in *Proceedings of the 2nd Joint Conference of the Information Retrieval Communities in Europe (CIRCLE 2022), Samatan, Gers, France, July 4-7, 2022*, L. Tamine, E. Amigó e J. Mothe, cur., ser. CEUR Workshop Proceedings, vol. 3178, CEUR-WS.org, 2022. indirizzo: https://ceur-ws.org/Vol-3178/CIRCLE_2022_paper_33.pdf.

[54] A. Cammozzo, E. Di Buccio e F. Neresini, «Monitoring Technoscientific Issues in the News,» in *ECML PKDD 2020 Workshops - Workshops of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2020): So-Good 2020, PDFL 2020, MLCS 2020, NFMCP 2020, DINA 2020, EDML 2020, XKDD 2020 and INRA 2020, Ghent, Belgium, September 14-18, 2020, Proceedings*, I. Koprinska, M. Kamp, A. Appice et al., cur., ser. Communications in Computer and Information Science, vol. 1323, Springer, 2020, pp. 536–553. doi: 10.1007/978-3-030-65965-3_37. indirizzo: https://doi.org/10.1007/978-3-030-65965-3_37.

[55] A. Minini, https://www.andreaminini.com/ir/stemming/.

[56] MathWorks, https://it.mathworks.com/discovery/stemming.html.